Project number: 653454
H2020-DS-2014-1

Project start: October 1, 2015
Project duration: 3 years

CREDENTIAL

D6.1

Pilot Use Case Specification

Document Identification	
Due date	December 31, 2016
Submission date	December 31, 2016
Revision	1.0

Related WP	WP6	Dissemination Level	PU
Lead Participant	FOKUS	Lead Author	Florian Thiemer
Contributing Beneficiaries	AIT, LISPA, ICERT, ATOS, OTE, KGH, FOKUS	Related Deliverables	D2.1, D6.4, D6.5, D6.6

Abstract: This document is dedicated to the specification of the pilots implemented within CREDENTIAL. It provides a detailed analysis and technical description of the selected use cases. Furthermore, it and describes how the two piloting phases will be organized in terms of pilot execution and evaluation.

This document is issued within the CREDENTIAL project. This project has received funding from the European Union's Horizon 2020 Programme under grant agreement no. 653454.

The information in this document is provided as is, and no warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Executive Summary

The CREDENTIAL Wallet is a cloud service hosted set of security and application services. These services provide authentication and authorization mechanisms combined with novel cryptographic technologies like proxy-re-encryption[1] and malleable signatures[2]. To showcase the functionality of the CREDENTIAL Wallet and to demonstrate how higher security and privacy levels can be achieved by the means of the CREDENTIAL Wallet, three different pilots in the domains of eGovernment, eHealth, and eBusiness are being developed. This document describes the execution plan of the pilots and defines quantitative and qualitative criteria to measure the success of each pilot. Furthermore, the document contains a detailed description of the technical use cases underlying the pilots, specified using UML sequence diagrams.

Slightly more explicit, each pilot will be executed according to a two-phase approach. The first phase will allow us to target early flaws and challenges, which will be addressed in a sanitization period between the two piloting phases. In the second phase, a more mature and more stable version of the pilots will be tested, potentially also offering a broader set of features to the users. Among others, this document details the execution of the two phases for the individual pilots, including practical aspects like the size and recruitment of user groups.

In particular, the results of the second phase will be used to evaluate the pilots' success. To do so, this document defines expected results and measurable key performance indicators which will be monitored during the pilot execution.

Finally, the document refines the business and logical use cases from D2.1 “Use Cases & Scenarios” by describing the technical aspects of the integration between pilot specific technologies and CREDENTIAL Wallet services. Thus, technical use cases in form of UML sequence diagrams were designed for each pilot. The CREDENTIAL Wallet architecture was developed in parallel to the work in this document leading to the question how and which components of the CREDENTIAL Wallet architecture are used by each pilot. The interaction between CREDENTIAL Wallet services and pilot specific technologies are described accordingly.

The results of this document build the frame for the development and implementation of the pilots towards the start of the piloting phase. Thus, it makes clear which parts of the CREDENTIAL Wallet are used by each of the pilots. This gives a better understanding which functionality of the CREDENTIAL Wallet needs to be implemented and how to integrate them with pilot-specific technologies. The pilots have a clear understanding which key performance indicators to keep track of and how to take actions within the pilot in order to measure them.

[1] M. Blaze, G. Bleumer, and M. Strauss, “Divertible protocols and atomic proxy cryptography,” in In EUROCRYPT. Springer-Verlag, 1998, pp. 127–144.

[2] R. Johnson, D. Molnar, D. Song, and D. Wagner, “Homomorphic Signature Schemes,” in Topics in Cryptology - CT-RSA 2002, vol. 28913, 2002, pp. 244–262

Document information

Contributors

Name	Partner
Nikolas Bompetsis	OTE
Jörg Caumanns	FOKUS
Pasquale Chiaro	ICERT
Enrico Francescato	ICERT
Agi Karyda	AIT
Alexandros Kostopoulos	OTE
Stephan Krenn	AIT
Andrea Migliavacca	LISPA
Juan Carlos Perez Baun	ATOS
Luigi Rizzo	ICERT
Anna Schmaus-Klughammer	KGH
Evangelos Sfakianakis	OTE
Florian Thiemer	FOKUS
Alberto Zanini	LISPA

Reviewers

Name	Partner
Stephan Krenn	AIT
Anna Schmaus-Klughammer	KGH
Fatbardh Veseli	GUF

History

0.1	2016-05-02	Florian Thiemer	Added Table of Contents and document structure
0.2	2016-05-10	Andrea Migliavacca	Added eGovernment LUCs
0.3	2016-05-13	Florian Thiemer	Added eHealth LUCs
0.4	2016-05-13	Pasquale Chiaro	Added eBusiness LUCs
0.5	2016-05-15	Nikolas Bompetsis	Modified eBusiness LUCs
0.6	2016-11-15	Juan Carlos Perez Baun	Added CREDENTIAL architecture chapter
0.7	2016-11-25	Evangelos Sfakianakis	Added generic Pilot Execution chapter
0.8	2016-11-28	Jörg Caumanns	Added eHealth Pilot technical description and TUCs
0.9	2016-11-28	Alberto Zanini	Added eGovernment Pilot description, Pilot Execution Plan, KPIs, TUCs
0.10	2016-11-29	Enrico Francescato	Added eBusiness Pilot description, Selected Use Cases, Pilot Execution Plan, KPIs, TUCs

0.11	2016-11-30	Alexandros Kostopoulos	Added Introduction, Abstract
0.12	2016-12-01	Stephan Krenn Agi Karyda	Reformatting to new template, moved TUCs and LUCs to Appendix
0.13	2016-12-04	Anna Schmaus-Klughammer	Added eHealth Pilot Execution Plan, Use Case Description, Expected Results, KPIs
0.14	2016-12-07	Stephan Krenn	Added list of abbreviations, started review
0.15	2016-12-11	Florian Thiemer	Added Execution Summary, Conclusion
0.16	2016-12-13	Florian Thiemer	Included readable summaries and introductions to eHealth LUCs and TUCs
0.17	2016-12-16	Florian Thiemer	Unified presentation and structure of the three pilot sections
0.18	2016-12-21	Stephan Krenn Juan Carlos Perez Baun	Revised abstract, executive summary, intro, methodology, and architecture
0.19	2016-12-23	Stephan Krenn Evangelos Sfakianakis Florian Thiemer	Restructuring of Appendix; fixed inner-document references; added introductions to various subsections
0.20	2016-12-27	Enrico Francescato	Revised details of eBusiness UCs and descriptions
0.21	2016-12-30	Luigi Rizzo	Final polishing of eBusiness description
0.22	2016-12-30	Stephan Krenn	Unified formatting, etc.
1.0	2016-12-31	Stephan Krenn	Final copy-editing

Table of Contents

List of Figures

List of Acronyms

2FA (3FA)	2 (3) Factor Authentication
APPC	Advanced Patient Privacy Consent
ATNA	Audit Trail and Node Authentication
BUC	Business Use Case
CEO	Chief Executive Officer
CGM	Continious Glucose Monitor
CNS	Carta Nazionale dei Servizi
CSP	Crypto Service Provider
DAC	Discretionary Access Control
DSUB	Document Metadata Subscription
eIDAS	Electronic Identification and Trust Services for Electronic Transactions in the Internal Market
FHIR	Fast Healthcare Interoperability Resource
HL7	Health Level 7
IHE	Integrating the Healthcare Enterprise
IdP	Identity Provider
IdPC	Identity Provider Cittadino
KPI	Key Performance Indicator
LDAP	Lightweight Directory Access Protocol
LOINC	Logical Observation Identifiers Names and Codes
LUC	Logical Use Case
PC	Personal Computer
PEC	Posta Elettronica Certificata
PHR	Personal Health Record
PKCS	Public-Key Cryptographic Standards
RBAC	Role-Based Access Control
SAML	Security Assertion Markup Language
SOAP	Simple Object Access Protocol
SFM	Service Functional Model
SIAGE	Sistema Agevolazioni
SMTP	Simple Mail Transfer Protocol
SP	Service Provider
SSL	Secure Sockets Layer
SSO	Single Sign-On
STORK	Secure Identity Across Borders Linked
T2DM	Type 2 Diabetes Mellitus
TLS	Transport Layer Security
TUC	Technical Use Case
UC	Use Case
UCUM	Unified Code for Units of Measure
UML	Unified Modeling Language
WP	Work Package
XDS	Cross-Enterprise Document Sharing
XML	Extensible Markup Language
XUA	Cross-Enterprise User Assertion

1. Introduction

With increasing mobility and Internet usage, the demand for digital services increases and has reached critical and high assurance domains like eGovernment, eHealth, and eBusiness. Those domains often have higher security and privacy requirements than other domains, and hence need to be harnessed with various novel mechanisms for secure access. Approaches for handling the resulting variety of authentication and authorization mechanisms include the use of digital identity and access management systems (IAM). Like other technologies, IAMs follow the trend of using cloud services. This allows abstracting over used resources and enables ubiquitous access to identity data which is stored and processed in the cloud, but also results in an additional degree of complexity for securely operating IAMs.

The goal of CREDENTIAL is to develop, test, and showcase innovative cloud based services for storing, managing, and sharing digital identity information and other potentially sensitive data. The security of these services relies on the combination of strong hardware-based multi-factor authentication with end-to-end encryption. The use of sophisticated cryptographic schemes enables a secure and privacy-preserving cloud-based information sharing network, in which even the identity and storage providers cannot access the data in plain-text.

The CREDENTIAL Wallet is the central component of the tools and components developed within the project. It offers a set of security and application services providing, among others, authentication and authorization mechanisms combined with novel cryptographic technologies like proxy-re-encryption and malleable signatures. To showcase the functionality of the CREDENTIAL Wallet and to demonstrate how a higher security and privacy can be achieved by the means of the CREDENTIAL Wallet, three different pilots in the domains eGovernment, eHealth and eBusiness target these aspects from different angles.

1.1 Scope

This document is dedicated to detailing the pilot scenarios and use cases to be implemented and executed. Specifically, the purpose of this document is threefold:

- The document develops and formally specifies the technical use cases required for realizing the intended pilots. Therefore, the selected use cases are described in detail, the used existing technologies, and also the pilot architecture and its integration with the central CREDENTIAL Wallet is described.
- Furthermore, it defines a clear description of how the different pilots will be executed. Besides a description how test users will be recruited or how their feedback will be incorporated, also a tentative time plan is provided.
- Finally, the document provides means to verify the success of the different pilots. For this, each pilot describes scope- and application-specific key performance indicators (KPIs), along with evaluation criteria and measureable success criteria.

1.2 Relation to Other Deliverables within CREDENTIAL

This document has incoming and outgoing dependencies with the following other deliverables of the CREDENTIAL project:

- **D2.1 “Scenarios and Use Cases”:** This document contained a detailed specification of business (BUCs) and logical use cases (LUCs) for multiple possible scenarios and use cases for our pilots. In the document at hand, we refine the BUCs and LUCs for the selected use cases by describing the technical aspects of the integration between pilot specific technologies and CREDENTIAL Wallet services. Thus, technical use cases (TUCs) in form of UML sequence diagrams were derived for the BUCs and LUCs from D2.1.
- **D5.1 “Functional Design”:** The CREDENTIAL Wallet architecture was developed in parallel to the work in this document leading to the question how and which components of the CREDENTIAL Wallet architecture are used by each pilot. This information was in particular required when formally specifying the TUCs mentioned above.
- **D6.4 – D6.6 “Test and Evaluation Report of Pilot Domain”:** These reports will contain detailed reports of the pilots. In particular, the evaluation will be based on the KPIs defined for each pilot in the report at hand.
- D2.5 “System Security Requirements, Risk and Threat Analysis – 2nd Iteration” and D2.6 “User Centric Privacy and Usability Requirements”: Those two requirements catalogues will, besides generic security, privacy, and usability requirements for the central CREDENTIAL Wallet, also contain scope-specific requirements for the different pilot scenarios detailed in this report.

1.3 Document Outline

The deliverable is organized as follows;

Section 2 presents the methodology we used with respect to the use cases specification, the pilots execution, as well as their overall evaluation. Then, in **Section 3**, we provide a brief overview of the CREDENTIAL architecture.

The core of this document is given by **Sections 4**, **5**, and **6**, where we describe in detail the pilots for the domains of eGovernment, eHealth, and eBusiness. Each domain is described by considering its scope, the selected use cases, the expected results, its KPIs and evaluation criteria, the required technologies as well as the pilot executions.

Finally, we briefly conclude in **Section 7**.

The comprehensive **Appendices A**, **7.B**, and **7.C** include the logical and technical use cases for each domain, where the presented UML diagrams depict the information exchange and the interactions between the involved actors, as well as the technical components.

2. Methodology

As explained before, the purpose of this document is to derive the technical use cases for the selected scenarios per pilot, describe the pilot execution, and specify the evaluation criteria. This section now explains the methodology used for each of those three goals.

2.1 From Scenarios to Technical Use Cases

CREDENTIAL features three pilots in the domains of eGovernment, eHealth, and eBusiness. In order to understand their functioning and behaviour, a top down approach by analysing the scenarios is performed. Figure 1 explains the dependency and contents across the different types of use cases; for a detailed treatment of the scenarios, BUCs, and LUCs, we refer to D2.1 “Scenarios and Use Cases”.

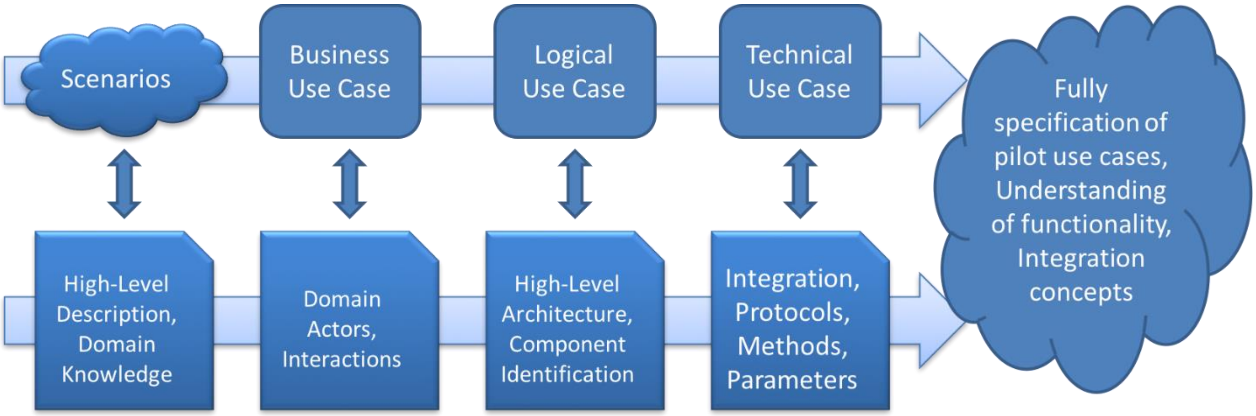

Scenarios
Business Use Case
Logical Use Case
Technical Use Case
Fully specification of pilot use cases, Understanding of functionality, Integration concepts
High-Level Description, Domain Knowledge
Domain Actors, Interactions
High-Level Architecture, Component Identification
Integration, Protocols, Methods, Parameters

Figure 1: CREDENTIAL use case specification approach

- **Scenarios:** The scenarios are the first starting point to explain the context of a certain problem in each of the three domains. By using personas and prose text, we describe which actors are involved, what special needs they have and what they are doing within this scenario. The scenarios describe in written form and using the domain’s terms and language what the pilot wants to do. Thus, we gain a high-level understanding of the domains.
- **Business Use Cases**: The next steps are the business use cases. These describe the pilots use case from a high level point of view without taking technical components or artifacts into consideration. The business use cases breaks down the scenario description in a more formal UML (Unified Modeling Language) notation in form of one or multiple sequence diagrams. Within the sequence diagrams the main actors from the scenarios and their interactions are described.
- **Logical Use Cases**: Based on the business use cases (BUCs), the logical use cases (LUCs) are derived. At this level, the focus is to have a clear understanding which high level architecture components are involved in the pilot. The actors are analyzed and components of the generic CREDENTIAL architecture identified within the logical use cases. Thus, the main interaction flow and sequencing between components are explained.

- **Technical Use Cases**: As a last step, the technical use cases (TUCs) describing those interactions in technical details are defined. The pilot environment and the CREDENTIAL architecture may use different kinds of protocols and standards. The technical use cases describe which technology is used and which operations and parameters are invoked between the components.

2.2 Pilot Verification

While the above artifacts' main purpose is to derive the CREDENTIAL architecture and give an understanding what the pilots want to achieve and how it could be technically done, it has still to be defined how the pilots are executed and how to determine if they have success. We therefore derive acceptence criteria and key performance indicators (KPIs) as follow:

- Each pilot defines expected results for its execution. They explain the motivation behind the execution of the pilot and explain where functional and non-functional gaps exist which can be closed by using CREDENTIAL technology. Each expected result is linked to one or more acceptance criterias. The acceptance criterias will be used within the evaluation phase to determine the success of the pilot and analyze which factors had a positive impact on the pilot, what may prevent the success, or what was missing during the execution as part of the lessons learned.
- In order to quantify the expected results each pilot defines multiple key perfomance indicators. For each KPI, this list contains a description, how the KPI is represented, how it will be measured, what is the reasoning behind it, how it can support the evaluation of the pilots, and which values or thresholds indicate a success or positive KPI.

2.3 Pilot Integration and Execution

Regarding the integration and testing/piloting, a two-phased approach will be applied for each of the pilots during their execution.

The methodology for the integration and testing is an agile testing approach as depicted in Figure 2. Briefly, this methodology considers testing and development as two intertwined phases and not sequential (i.e., testing following development). This decision was taken due to need for an early integration of all developed components to allow for an on-time start of first piloting phase, which will then provide feedback back to the developers.

With respect to the practical, separate, and integrated module testing, we opted for an approach proposed by Myers et al. [1]. Figure 2 shows a generic perspective of the integration, testing, and validation processes, which will be adopted by the CREDENTIAL team for the modules that will be integrated to produce the final CREDENTIAL Wallet (the numbers refer to steps described in the following):

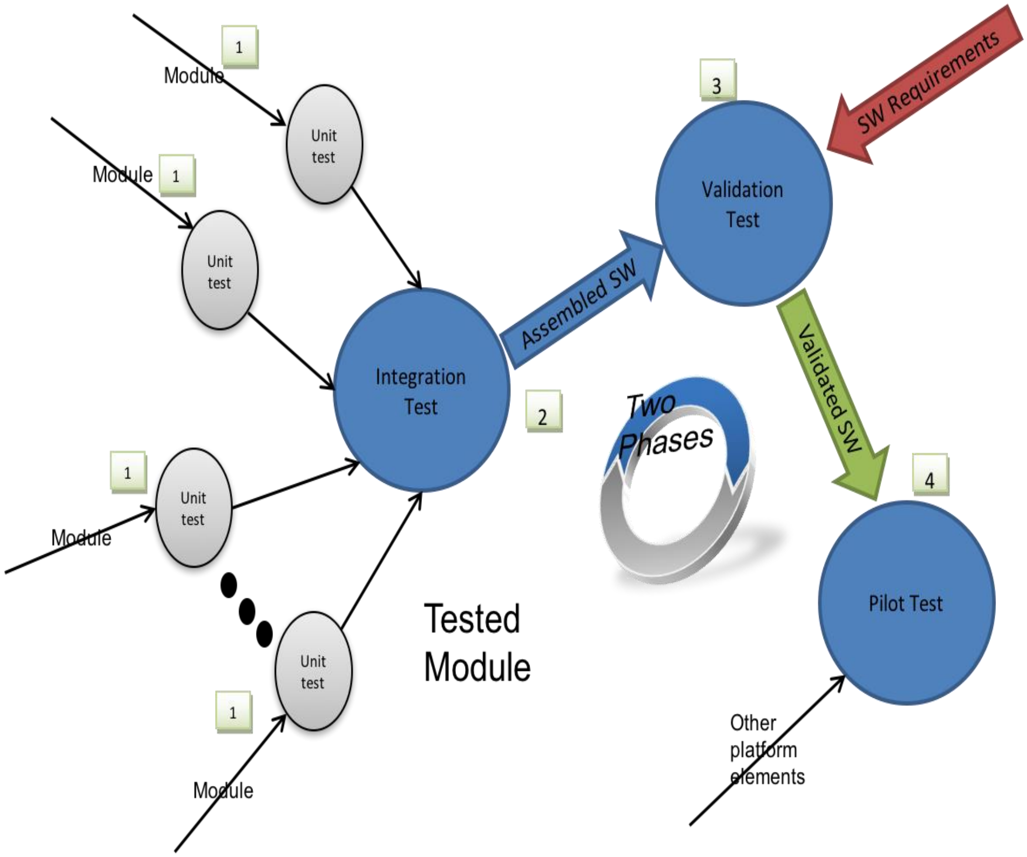


Figure 2: The general testing methodology

According to this generic methodology, our plan is composed of the following major steps:

1. Individual unit testing to ensure the correctness of single modules developed by the different partners.
2. Integration and subsequent testing of the individual units to implement the CREDENTIAL Wallet platform.
3. Validation testing of the integrated platform against the requirements specified in the respective CREDENTIAL deliverables.
4. Integration and integration testing of the CREDENTIAL Wallet platform with the different environments from the three pilot domains.
5. A first pilot phase of the different pilots integrated with the CREDENTIAL Wallet will allow us to target early flaws and challenges.
6. Feedback from the first pilot phase is given to developers and corrective measures are implemented whenever necessary in this sanitization period. The updated modules will be unit-tested against the reported problems similar to Steps 1 and 2.
7. Integration of new individual unit modules addressing feedback from the first piloting phase and/or offering extended functionality for the pilots.
8. Similar to Step 4, the fully integrated pilots will be tested internally.
9. The second phase of the pilot will be used to collect feedback for evaluating the success of the three different CREDENTIAL pilots according to the criteria defined in this document.

In Figure 3 we see the testing process for each integrated module:

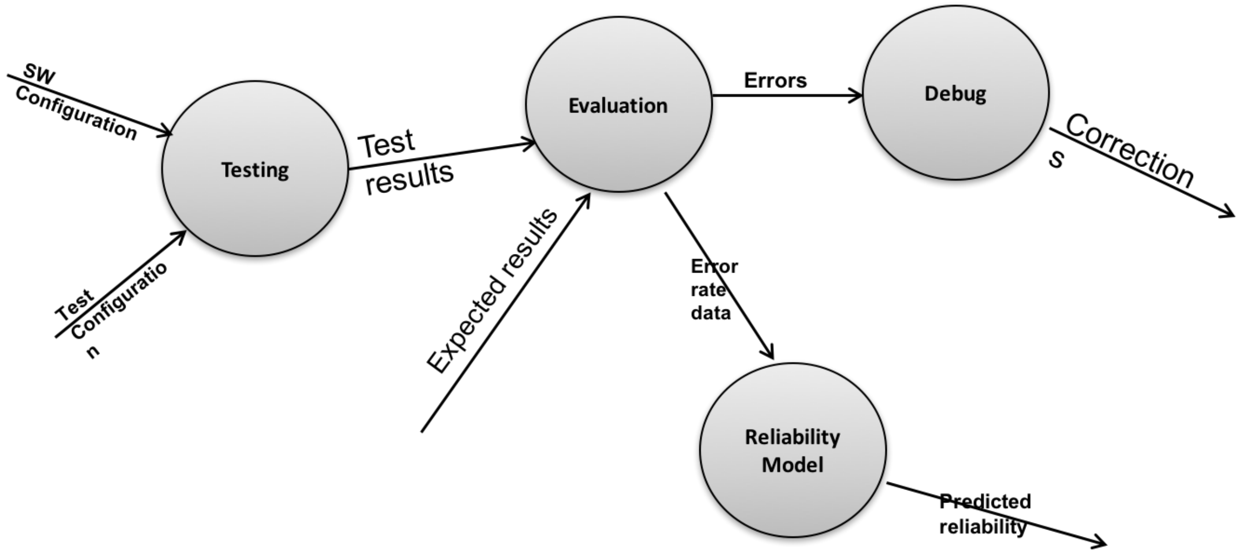


Figure 3: Testing a module

With respect to system testing, there are generally two main methodologies: Incremental testing, and Non-incremental testing. We, briefly, discuss these two methodologies in what follows and state, with justification, our adoption decision. In the context of our description, the term *low-level component* refers to self-contained modules which perform a specific, relatively simple, computation. The term *high-level component* refers to modules which perform more complex operations and require other modules for their operation.

1) Incremental system integration

- *Top-Down testing*: This approach requires the integration and testing to start from the highest-level components. This enables the testing of high-level system parts and the involved data flows and interfaces. This approach, *usually*, minimizes the need for *drivers*, i.e., test modules that invoke others, since most modules are in place (at least the high-level ones). However, since lower level modules are missing, *stubs* are necessary (perhaps numerous) to feed with inputs the higher-level component which are under testing, until the real lower level components become available. In addition, the lower level components are tested late in the integration phase leaving limited time for system level corrective actions. The exact form and functionality of the stubs and drivers, since they form very low code-level components of the testing phase, depend heavily on the actual code of the tested modules, which is not available at the time of the writing of this deliverable, will be decided later by the involved partners.
- *Bottom-Up testing*: Contrary to top-down testing, this testing strategy starts from the lower level system components. This, also, minimizes the need of stubs and identifies low level problems early, before the integration begins. However, more drivers are needed to invoke, appropriately, the lower level components because the higher level components (which invoke the lower level ones) are missing.

- *Sandwich Testing*: This is also called *Hybrid Integration Testing*, and combines the two approaches discussed above. According to this approach, lower level modules are tested in parallel, while their integration also proceeds along with testing and is tested itself too. Accordingly, the need for stubs and drivers is minimized. However, this approach is a little less systematic than the top-down and bottom-up approaches and may need more coordination effort.

2) Non-Incremental system testing:

- *Big-bang testing*: In this approach, all (or a large portion) of the modules that compose the system are integrated and tested. According to this testing methodology, the testing actually starts from the final system and not the individual components level. However, in this testing strategy, if an erroneous behavior is detected while a test is performed, it is very difficult to isolate the components that fail and lead to this kind of behavior, since attention is not paid at the component operation or component interface level during the testing.

After consideration of these options, the development teams are opting for the “sandwich approach”. This methodology is more suitable due to the fact that partners, in parallel, develop their own modules according to the specifications set in the architecture deliverables (most notably D5.1). Thus, the module developers use the bottom-up approach to test their low-level modules and then perform an integration with the reference architecture as a result of “D5.6 – Reference Environment” and the CREDENTIAL Wallet hosting and pilot set-up as a result of “D6.2 Identity wallet services” apply the bottom-up approach and support the piloting phase.

Pilot evaluation happens after the execution in each phase for each pilot separatley. The evaluation has two goals:

- Identify bugs within the implementation and adopt the functionality according to user feedback which is given in the first execution phase.
- Determine the success of the pilot by matching the results to predefined expectations after the second execution phase.

The results after the first execution phase will be collected within a lessons learned catalogue for each pilot. It contains the user feedback collected during the first phase. According to these feedbacks each pilot individually analyse what was satisfying for the users, what was unsatisfying and which features may be added in order to improve the functionality and usability of the pilots. These results are bundled together with a detailed bug and test report by the pilot partners. The main goal is to keep the pilot on track and identify problems during the execution thus that the second phase can be performed under optimal conditions.

The second phase of the pilot is performed with adjustment made according to the results in the lessons learned catalogue. Here the pilot is finally evaluated and the success of the pilots are determined.

The evaluation is supported by a set of key performance indicator (KPI) and by a list of expected results defined by each pilot partner individually. The KPIs describe:

- What will be measured?

- How will it be measured?
- How is the value represented?
- What are the successful thresholds for the KPI?

The KPIs supports the evaluation process in order to give quantitative and qualitative criterias and measurements for the pilot partners. These KPIs will be measured during each of the pilot test phase and the results presented and anylized in the evaluation reports. Furthermore, by comparing the KPIs between the two phases an improvement of the pilot in the second phase can be measured.

The expected results are high level description of successful criterias for the pilots. KPIs are related to the expected results and help to determine a successful or not successful outcome of the expected result.

The evaluation results of the lessons learned catalogue and the pilot evaluation report will be released together with the final results in the upcoming deliverables D6.4-D6.6 (Test and evaluation reports) for each of the pilots.

3. CREDENTIAL Architecture

This section introduces the CREDENTIAL architecture derived from D5.1 and describes how the selected pilot UCs are covering some of the components the architecture provides.

The main actors for a CREDENTIAL ecosystem are:

- ***CREDENTIAL Wallet***: The CREDENTIAL Wallet stores user data and identity data in a secure cloud. It is a cloud platform, which offers sharing of those user data with other participants or service provider in a secure way and preserving user privacy. The Wallet comprises an *Identity and Access Management* system, performing authentication and providing authorization to access those data. An external Identity Provider can be embedded to offer authentication functionality for end users.
- ***CREDENTIAL Participant***: A CREDENTIAL Participant is any member interacting with the CREDENTIAL Wallet. Several types of participants can be distinguished:
  - ***End Users:*** The user stores its data in the CREDENTIAL Wallet. It is the owner of the data in the cloud and has the absolute control over the data flow of its personal and sensitive data.
  - ***Service Provider***: The Service Provider acts like data receiver, which offers domain specific functionality for other users.

Three main categories of functionalities are provided by the CREDENTIAL platform:

1. The ***Account Management*** services focus on the whole account life-cycle and access management. A user can create a new account that involves the creation of its proxy-re-encryption enabled key material and an account association on the CREDENTIAL Wallet. Furthermore, the user can perform various management functionalities like showing an activity protocol on its data or delegate access rights to its data.
2. The ***Identity Management*** functionalities are focusing on integrating identity data stored within the CREDENTIAL Wallet in the authentication mechanisms towards other service providers. The use of proxy-re-encryption technologies allows sharing the identity data in the CREDENTIAL Wallet in a secure and privacy aware way.
3. The ***Data Sharing*** services focus on storing, reading and sharing of user data that is assigned to the CREDENTIAL Wallet. The user data is protected by encryption and sharing of the user data is never disclosed to the CREDENTIAL Wallet itself.

Figure 4 depicts the layered architecture of CREDENTIAL. This figure shows the major actors and components involved to provide the above described functionalities. More information about the CREDENTIAL architecture can be found at D5.1 “Functional Design”. Based on this overview of the CREDENTIAL architecture, the remainder of this section will explain how these components are used to implement the individual pilots.

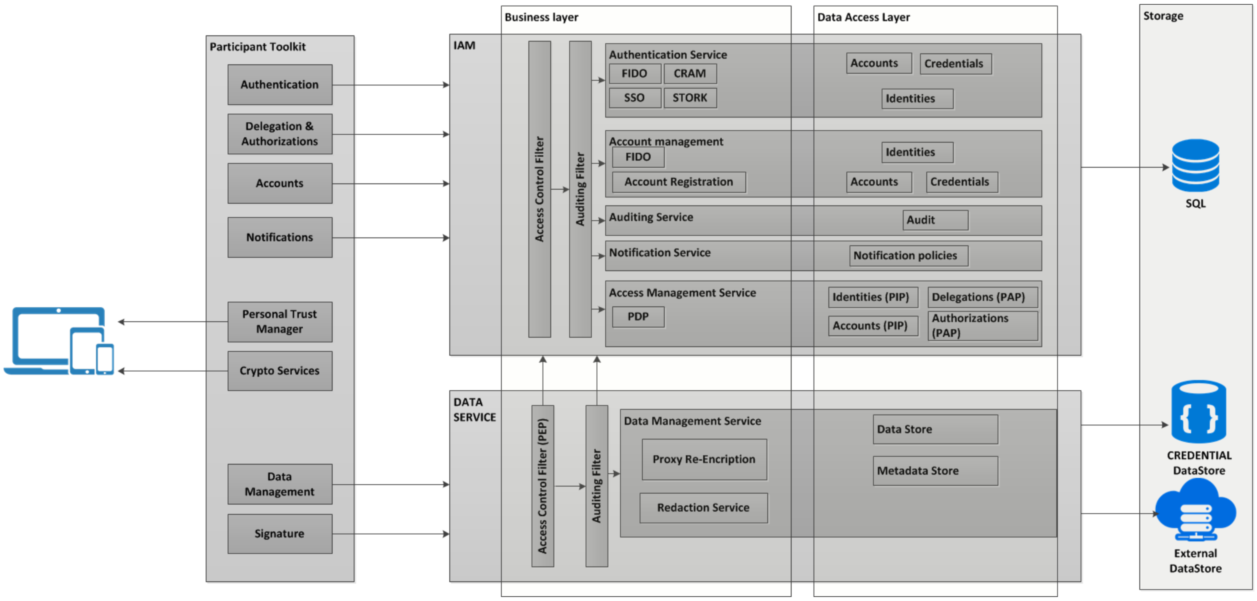


Figure 4: CREDENTIAL architecture in layers

CREDENTIAL will be integrated in three pilots covering the eGovernment, eHealth and eBusiness domains. The eGovernment pilot is focussing on Identity Management, the eHealth pilot targets sharing of sensitive medical data across multiple users, and the eBusiness pilot looks for sharing of data with service providers.

However, the above generic functionalities provided by CREDENTIAL are going to be invoked in the pilots in one way or another:

- Account management: you need a CREDENTIAL account first!
- Identity Management: you are asserted.
- Data Sharing: exchanging data with CREDENTIAL Wallet.

3.1 eGovernment Pilot using CREDENTIAL

The eGovernment pilot is focussing on Identity Management. LISPA has selected two UCs to be implemented within the CREDENTIAL platform. These UCs are based on the user access to a service (SIAGE) provided by the Lombardy region.

- For this purpose, an Italian citizen needs to gain authentication interacting with certain Identity Provider. The Italian citizen will get the authentication interacting with the Lombardy IdPC, which should access to CREDENTIAL Wallet to enrich the user assertion.
- For those non-Italian European citizens trying to access the Lombardian service, a component called “IdP adapter” calls CREDENTIAL Wallet to get user identity data or assertion using the eIDAS infrastructure. The European citizen will use the CREDENTIAL mobile app for a strong and secure authentication.

The CREDENTIAL components to be integrated with in the above UCs are identified in Figure 5. Red boxes indicate the components embedded into the CREDENTIAL Participant Toolkit, which provides user's access to CREDENTIAL Wallet through different devices (smart phone, tablet or laptop). Taking into account the eGovernment UCs, described more extensively in Section 4, the Participant Toolkit components are linked by red arrows to the services offered by CREDENTIAL Wallet. Orange boxes are framed the services used by the indicated UCs in both the business layer and the data access layer.

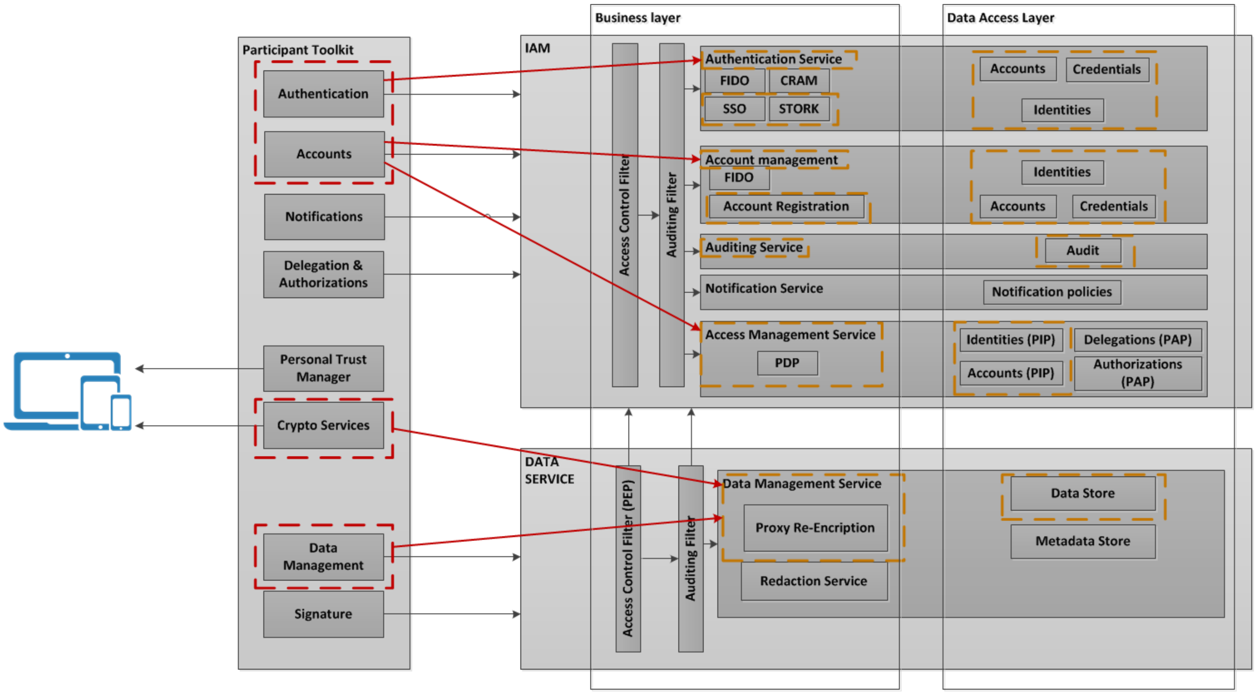


Figure 5: eGovernment pilot in CREDENTIAL

In both use cases, the citizen owns a CREDENTIAL account, and manages this account through the Account Management service. Also, CREDENTIAL cryptographic libraries will be used in order to encrypt and decrypt user identity data. The authentication service is involved either when the Lombardy IdPC or the eIDAS infrastructure is required.

For the second UC notifications to the user's mobile phone are pushed from the notification service during the authentication process, also the Authentication service subcomponents such as STORK/eIDAS will be used for cross-border authentication purposes.

Auditing service and Access Management service as generic services are involved in all the process.

3.2 eHealth Pilot Using CREDENTIAL

The eHealth pilot is addressed for sharing of sensitive medical data across multiple users. The aim of this pilot is demonstrating how CREDENTIAL privacy and security technologies can be used in conjunction with established eHealth standards. By this the pilot will show how

- CREDENTIAL services are integrated with existing health information exchange networks and as such add full patient control to existing solutions,
- established eHealth standards and profiles can interplay with CREDENTIAL with full conformance to the standards' logical and technical specifications,
- how CREDENTIAL services can be integrated as a re-usable security architecture layer which can be loosely coupled to arbitrary business service layers.

For doing so, the CREDENTIAL eHealth Pilot utilizes an off-the-shelf open source Health Record solution for implementing a Personal Health Record (PHR). All documents are stored and shared in encrypted manner only with the CREDENTIAL Wallet being used for securely sharing encryption keys between the patient and his doctors. In addition, all audit trail entries generated by the PHR are securely stored to the CREDENTIAL Wallet.

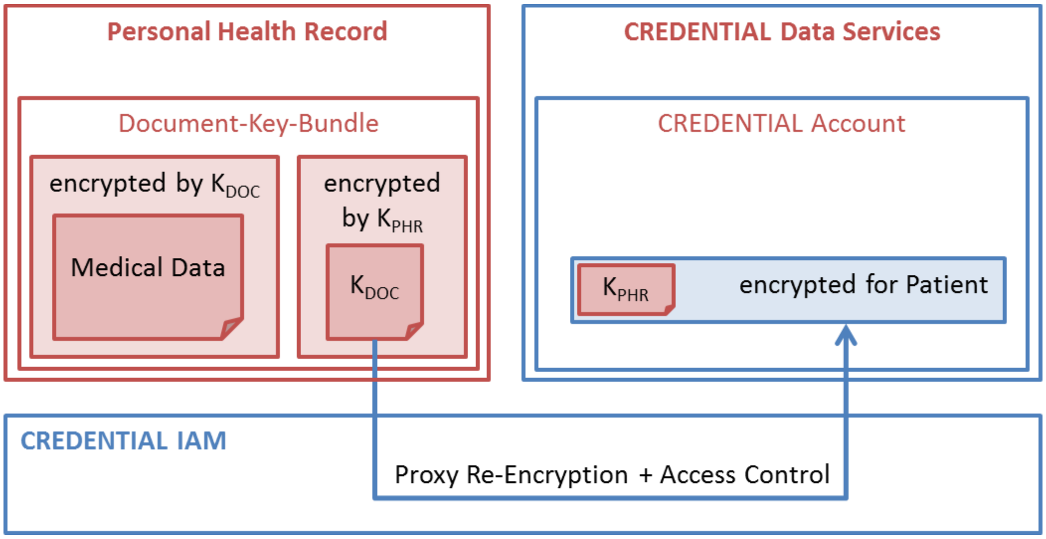


Figure 6: End-to-End Encryption

Figure 6 sketches how the introduction of a PHR-Key allows for providing end-to-end security without being forced to break existing healthcare-IT standards:

- A dedicated PHR-Key (Key_{PHR}) is generated for each instance of a Personal Health Record.
- Using hybrid encryption, each medical document is encrypted with a dedicated key (Key_{DOC})
- The document key Key_{DOC} is encrypted by the PHR-Key of the PHR that document is assigned to.
- By treating PHR-Keys as documents, the CREDENTIAL Wallet is used for securely sharing PHR-Keys between the patient and his doctors.
- If a doctor is granted access to a patient's PHR, CREDENTIAL proxy re-encryption is used for re-encrypting the PHR-key for the authorized doctor.

The UCs selected by the eHealth pilot are related to access and sharing medical data among different health stakeholders in a secure and privacy way, and are described in deep in Section 5:

- **Care Planning and Progress Tracking:** Patient and doctor own CREDENTIAL *accounts* and the PHR is *stored* in the CREDENTIAL Wallet. Such information is shared after patient and doctor agreement.
- **Nutrition and Activity:** Patient uploads to CREDENTIAL Wallet this information and shares it with doctors.
- **Therapy Monitoring and Screening for Complications:** Notifications are sent to the doctor by the CREDENTIAL Wallet, which checks the correspondent thresholds in order to push the notifications.

Notice in Figure 7, that the eHealth pilot covers every function of the CREDENTIAL platform.

As indicated in the previous section the red boxes indicate the components embedded into the CREDENTIAL Participant Toolkit, these components are linked by red arrows to the services offered by CREDENTIAL Wallet and the orange boxes are framed the services used by the eHealth UCs in both the business layer and the data access layer.

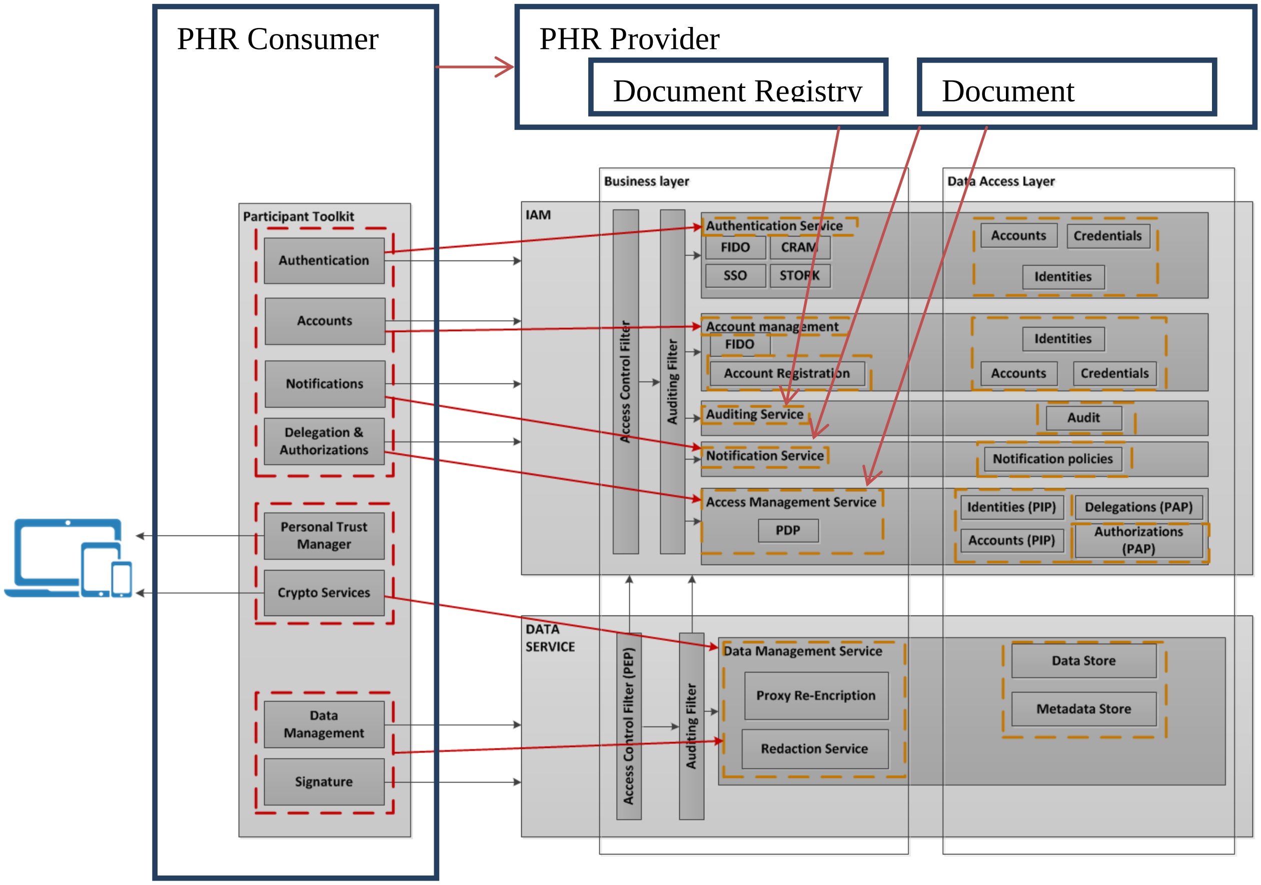


Figure 7: eHealth pilot in CREDENTIAL

Patients and doctors own CREDENTIAL *accounts* and manage these accounts through the CREDENTIAL eHealth App. By using the CREDENTIAL account identifier as the PHR patient identifier, a 1:1 correspondence is established which leads to the notion that a patient’s PHR is set up within his CREDENTIAL account.

Authentication and authorization processes for accessing PHR are provided through CREDENTIAL security libraries and services. Notifications are triggered by the PHR and utilize defined CREDENTIAL means for receiver discovery and message delivery.

When the patient uploads medical data and shares this information with a doctor the Data management service is involved.

The patient thanks to the auditing service can monitor who and when is accessing her data.

3.3 eBusiness Pilot Using CREDENTIAL

The eBusiness pilot looks for sharing of data with service providers. InfoCert has chosen the following UCs, which are widely specified later in this document in Section 6:

- **InfoCert e-commerce login with CREDENTIAL**: *e-commerce login (web-based application)*
  The user will be able to login to the InfoCert e-commerce using its CREDENTIAL Wallet. Some data must be imported to let the InfoCert e-commerce create a user account and bind it to the user’s wallet.
- **Legalmail contract form filling with CREDENTIAL**: e-commerce form filling (*web-based application*)
  The user, after logging in with CREDENTIAL can request a Legalmail mailbox. Part of the contract form can be filled with *data imported* from the CREDENTIAL Wallet.
- Legalmail encrypted message forward (*mobile application*)
  The Android Legalmail App will request to the CREDENTIAL Android app to *generate a re-encryption key* and will send it to the Legalmail server, which will store it and use it, by means of the CREDENTIAL Java libraries, when a forward filter is activated.
  The user that will receive the forwarded message will use the Android Legalmail App. The Android Legalmail app will request to the CREDENTIAL Android app to decrypt the session key which, when returned, will be used to decrypt the original content (S/MIME protocol).
  This use case does not use the CREDENTIAL Wallet, but only the features provided by the CREDENTIAL mobile app and the CREDENTIAL Java libraries.

The CREDENTIAL components to be integrated with in the above UCs are identified in Figure 8.

As aforementioned the red boxes indicate the components embedded into the CREDENTIAL Participant Toolkit, The Participant Toolkit components are linked by red arrows to the services offered by CREDENTIAL Wallet and the orange boxes are framed the services used by the eBusiness UCs in both the business layer and the data access layer.

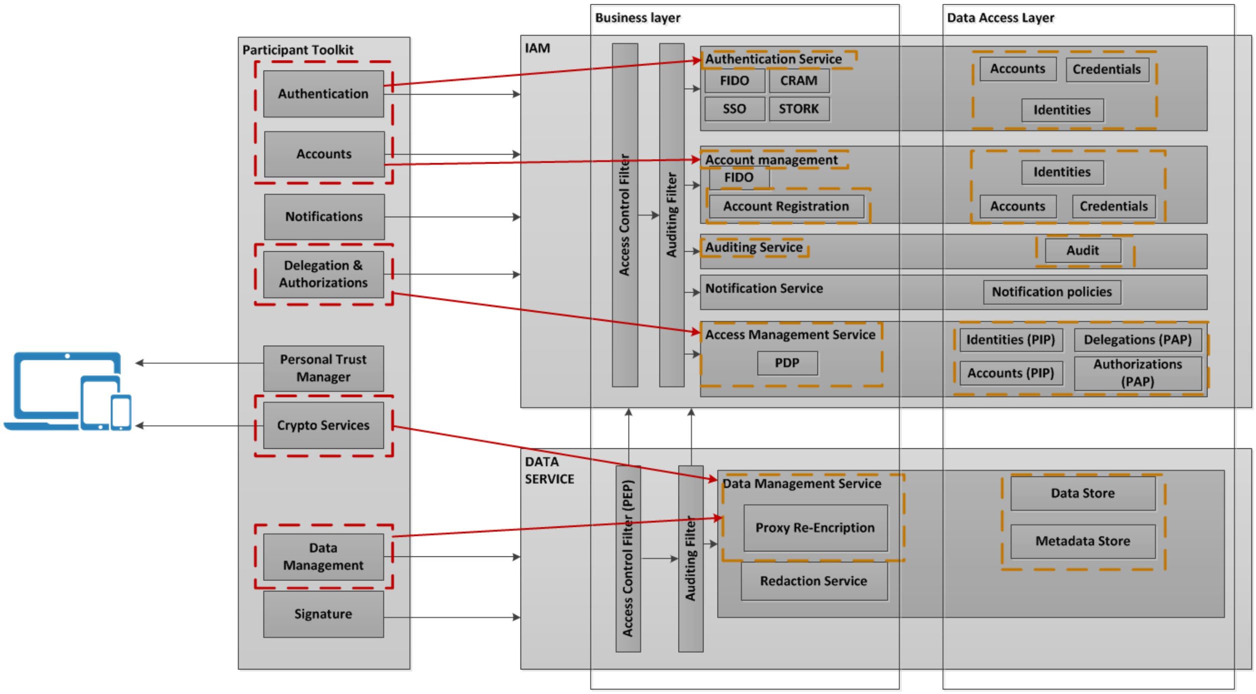


Figure 8: eBusiness pilot in CREDENTIAL

In all the UCs the user owns a CREDENTIAL *account*, and manages this account thanks to the Account Management service. Two of the UCs selected by InfoCert are related to the InfoCert e-commerce web based application involving the use of CREDENTIAL Wallet. Mainly the CREDENTIAL Wallet is used for authentication purposes, and for filling a contract form with *data imported* from the CREDENTIAL Wallet. For this purpose, the Access Management Service and the Data Management service are also involved. The third chosen UC employs the CREDENTIAL Wallet to authenticate the user for the Legalmail service. The fourth chosen UC, the Legalmail encrypted message forward, does not use the CREDENTIAL Wallet, but only the features provided by the CREDENTIAL mobile app and the CREDENTIAL Java libraries, namely the re-encryption mechanisms.

4. eGovernment Pilot

The eGovernment pilot consists of a web application available to citizens on the Internet which aims to simplify the relationship and interactions between users and the government. In this domain, Lombardian citizens have a wide portfolio of web applications offered by Lombardy Region and one of these is SIAGE, a web portal useful to request tax breaks and other types of fiscal advantages.

SIAGE and its user can have several benefits using CREDENTIAL components. In fact, CREDENTIAL provides a user-friendly and secure way to share eGovernment-related identity data, while the user stays in full control of her/his personal information.

The CREDENTIAL architecture is integrated for the following major goals:

- to assure confidentiality during secure user online authentication
- to demonstrate cross-border interoperability
- to securely get user data, stored in an encrypted form in the Wallet.

The benefits over the state of the art are:

- User data collected into an assertion, released by an Identity Provider, are encrypted; only target Service Provider can decrypt those values; this means that attacks like man-in-the-middle (or similar) are ineffective, and at the same time user is sure that her/his data can be accessed in plain form by Service Provider only (Identity Provider just collects encrypted data and forward them to Service Provider). This goal is possible thanks to proxy re-encryption technique.
- Any Identity Provider can be plugged into the CREDENTIAL architecture and can be used to release identity assertion; the only effort requested is to enhance a component, called "IdP Adapter", created to translate several authentication protocols.
- User can store into the Wallet additional data, over the pure identity data; those data benefits of the same encryption and CREDENTIAL components assure that only the target Service Provider can access to plain data.
- During authentication, user can apply a selective disclosure of her/his data, in order to forward to Service Provider just the personal data she/he wants.

The pilot will be set up in LISPA and other partners' environments. The evaluation will be performed by measuring technical values (technical KPIs) and interviewing users, in order to "measure" their grade of satisfaction.

This section will describe, with an appropriate level of detail, the scope and architecture of the pilot, the selected use cases (explaining which user cases will be implemented and the reason behind the choice), the expected results and the Key Performance Indexes chosen, the technologies involved in this pilot and a view of the pilot setup, including a high-level project time line.

4.1 Scope

The scope of this pilot is focused on eight goals, described in this section and depicted as follow.

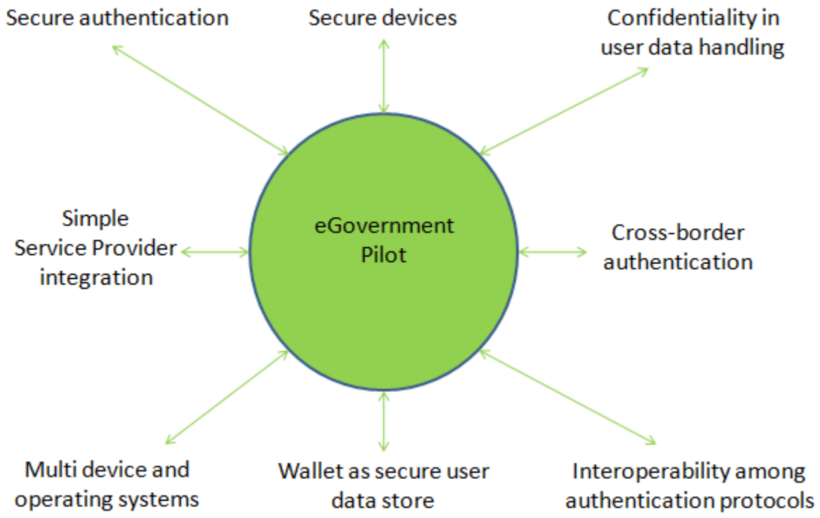

Secure authentication
Secure devices
Confidentiality in user data handling
Simple Service Provider integration
eGovernment Pilot
Cross-border authentication
Multi device and operating systems
Wallet as secure user data store
Interoperability among authentication protocols

Figure 9: eGovernment pilot scope and goals

The eGovernment pilot is mainly oriented on *secure authentication* in a cloud environment using an ecosystem of Identity Providers (IdP) which join the CREDENTIAL project. Secure authentication covers at least two key factors: user is authenticated in a secure way (using a *secure device*) and user data are handled and transmitted over the network with confidentiality and security.

Another important goal of the eGovernment pilot is to demonstrate that is possible, for a *Service Provider* (SP), to *integrate* CREDENTIAL in a *simple* way and with a limited effort. In Lombardy Region, this scenario would be at "zero cost", due to previous integration with the Lombardy Region Identity Provider (IdPC), already done by hundreds of SP.

A major goal of eGovernment pilot is to demonstrate *cross-border authentication*: an effective federation of several national IdPs is possible using eIDAS infrastructure, which is a modern, updated approach already experimented with STORK components. The eIDAS infrastructure aims to become a de-facto standard in authentication, and is a modern, updated approach compared to STORK ecosystem.

eGovernment pilot assumes that users could use a desktop PC to work with the SP and a mobile device to authenticate. This approach will demonstrate that a wide variety of *devices* and *operating systems* can be used to access CREDENTIAL components.

It is important to underline that the CREDENTIAL Wallet component can be also used as a secure *store of user data*, even if its main role in this pilot is indeed the Identity Provider. The Wallet can be accessed – in a secure way – by any IdP to retrieve additional user data, in encrypted form.

Another goal of the pilot is to demonstrate a full *interoperability* between different *authentication protocols* and mechanisms. The component called “IdP Adapter” plays a key role, given that its task is to translate and adapt the various authentication “languages” used in CREDENTIAL project.

4.2 Architecture Description

Actual architecture of the pilot consists of seven main components:

- A user agent (browser) typically running on a desktop PC or on a laptop - it is used by citizen to access to the SP
- A SP (namely SIAGE) – it is a web application, available on the Internet, which offers several economic advantages or tax breaks to certain categories of Lombardian citizens. SP will be hosted by LISPA
- A Shibboleth reverse proxy – its task is to redirect user browser to an IdP Selector when needed. This component will be hosted by LISPA
- An IdP Selector/Adapter – this component plays a double role: it offers a chooser of the IdPs available in CREDENTIAL, and translate/adapt authentication protocols when needed. This component will be hosted by LISPA
- The Lombardy Region IdPC (IdPC) – it offers secure authentication via an SSL/TLS connection established with the user browser using a smartcard with digital certificates on-board. This component will be hosted by LISPA
- The CREDENTIAL Wallet – it is mainly used as Identity Provider but it can also be used as a secure store of additional user data. CREDENTIAL Wallet will be hosted by OTE
- eIDAS Adapter – it is used when cross-border authentication is chosen by the user. The Adapter will be implemented by FICEP (First Italian Crossborder eIDAS Proxy).

4.3 Selected Use Cases

LISPA eGovernment pilot mainly consists of two use cases. A detailed description of both use cases is available in this section and also in the Appendix 0.

4.3.1 Lombardian Citizen Accesses eGovernment Service

In the first use case, described with UML diagrams at paragraph 3.1, a Lombardian citizen access to an eGovernment Service Provider (SIAGE) using her/his CNS smartcard as authentication mechanism. The Identity Provider involved in this use case is the Lombardian IdP, called IdPC, which offers several ways to authenticate in a secure way the user, mainly by a smartcard or by an OTP/SMS 2FA. CREDENTIAL cryptographic libraries assure confidentiality in this case. The CREDENTIAL Wallet may be used in order to retrieve additional user data if needed to a service provider. A second version of the same use case involves the usage of the eIDAS architecture: user authentication implies the choice of the same, domestic Lombardian IdP but passing through eIDAS nodes. This scenario is established to demonstrate eIDAS interoperability and cross-border authentication feasibility. If available at the date of pilot delivery, a foreign IdP joining the eIDAS network will be also used.

The flow of the first use case is described in Figure 10. It shows a user who uses a desktop Pc with a smartcard reader and a classic desktop browser. Every user interaction with the SP and the Identity

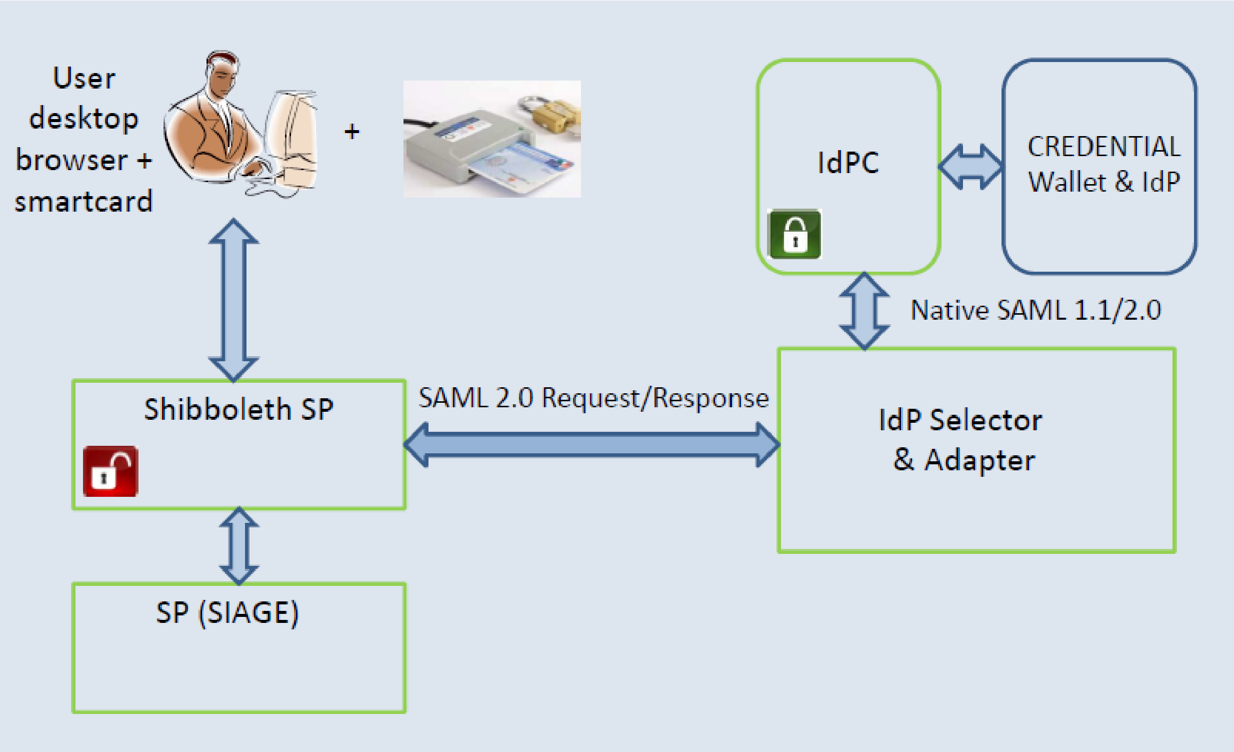


Figure 10: Interaction between Lombardy SIAGE services and the CREDENTIAL Wallet of the first eGovernment use case

Provider is done with that desktop browser. User browser is redirected to an “IdP Selector/Adapter” as soon as the user tries to access to a SP web page that requires authentication. These protected paths are configured in Shibboleth SP, who is also the component responsible for the browser redirection and the handling of SAML authentication request/response. The IdP chooses the domestic IdPC which creates a TLS/SSL mutual authentication with user browser – this implies that user is asked to insert smartcard PIN in order to electronically sign SSL handshake session. IdPC performs a deep analysis of user certificate – the X.509 certificate must be not revoked, not expired, and issued by a trusted Certification Authority. If all of the previous conditions are satisfied, IdPC releases a SAML authentication token (in 1.1 or 2.0 format) – the “IdP Adapter” is involved in the first case only if IdPC releases a SAML 1.1 assertion, due to protocol translation need. The user has the possibility to apply selective disclosure of her/his data before IdP prepares and forwards authentication assertion.

4.3.2 Foreign Citizen Accesses eGovernment Service

In the second use case, described with UML diagrams at paragraph 3.1, a non-Lombardian citizen (for example, a Spanish citizen) access to Lombardian SP SIAGE using CREDENTIAL mobile APP as a secure authentication mechanism. This implies the usage of CREDENTIAL Wallet IdP instead of Lombardian IdP. Even in this use case, CREDENTIAL cryptographic libraries are used to encrypt/decrypt identity data, using proxy re-encryption.

The flow of the second use case is described inFigure 11. The user now has two devices: a desktop PC with a classic browser to interact with the SP, and a mobile device to run CREDENTIAL App in order to authenticate herself/himself to Wallet – which runs here an IdP role. Lombardian IdPC has no role in this use case and is not depicted. The “IdP Adapter” is used in any case because the assertion released by

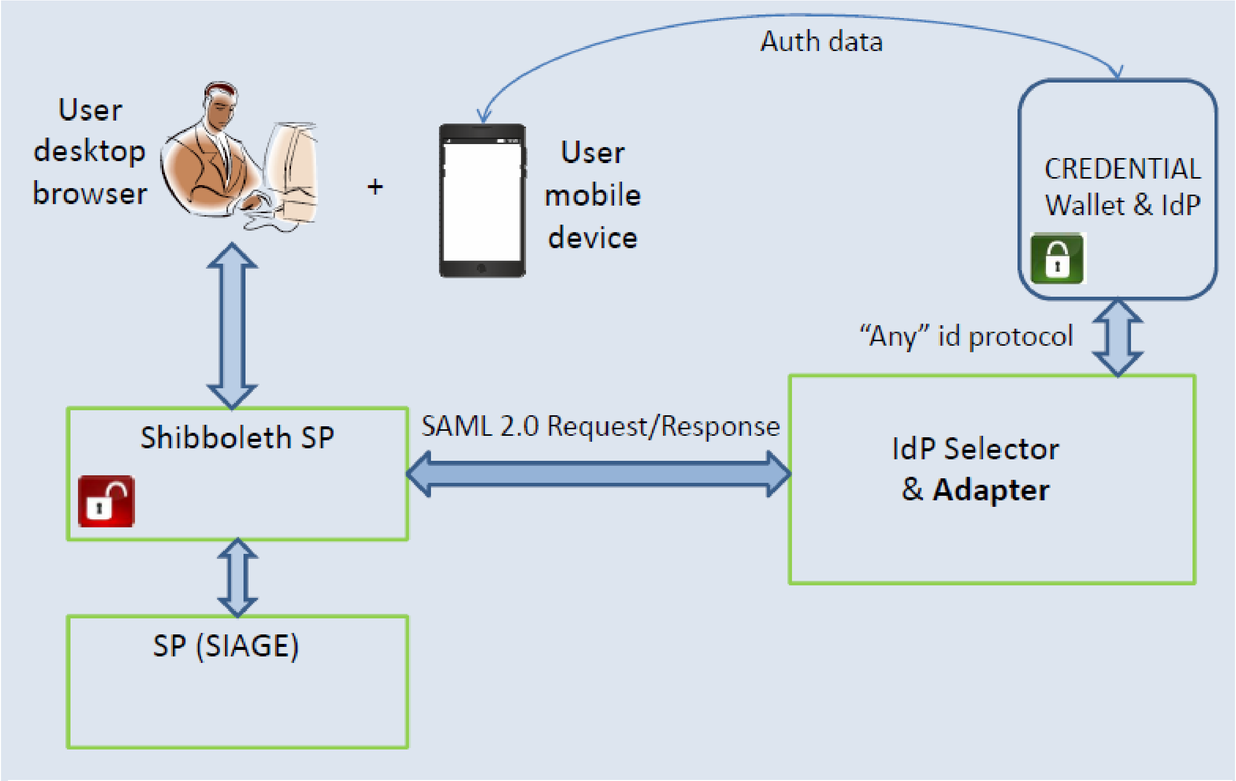


Figure 11: Interaction between Lombardy SIAGE services and the CREDENTIAL Wallet of the second eGovernment use case

CREDENTIAL Wallet IdP is certainly different from the one released by Lombardian IdPC, and a "translation" is needed in order to preserve SP. Even in this use case, the user has the possibility to apply selective disclosure of her/his data before IdP prepares and forwards authentication assertion.

4.3.3 Selection Justification

These use cases have been chosen for the following reasons:

- Usage of different Identity Provider is a good way to demonstrate that the CREDENTIAL architecture is designed to host different IdP, even with different authentication protocols. An "IdP Adapter" plays a key role in order to "translate" the original, domestic SAML 2.0 request/response into "any" target IdP protocol;
- The pilot architecture is explicitly designed to host "any" Lombardian SP without the need to modify them in any way. Even this capability is an added value of the "IdP Adapter": the SP continues to produce the usual authentication request and receive the usual authentication response, and it is not aware of the presence of non-domestic IdP (for example the CREDENTIAL Wallet IdP or a Spanish IdP);
- Every use case involves CREDENTIAL cryptographic libraries – the goal is to integrate them in at least two different IdP (IdPC and CREDENTIAL Wallet IdP) to demonstrate their flexibility;
- At least one use case (the second one) involves a non-Lombardian IdP – the goal is to demonstrate that the entire architecture is not "IdPC-dependent";
- At least one use case (the second one) is a good example of the "mobile first approach": user is asked to use her/his mobile device to have authentication granted;

- A third use case was formerly designed in order to have a third-party SP (for example a Spanish SP) – the Lombardian user authenticates herself/himself to domestic IdPC and access to a non-Lombardian SP in order to perform some operation. This use case has been deleted because the goal of have a demonstration of cross-border authentication is also covered developing the second "version" of the first use case, as mentioned above.

4.4 Expected Results

The expected results of this pilot mainly cover technical aspects, but in some cases also refer to a perceived added value by end users. For example, one of the main expected result is that user can access any SP service regardless to which authentication method she/he choose. Furthermore, user should find the use of a CREDENTIAL mobile app more attractive compared to the use of a smartcard. Therefore, another expected result is that some transactions during the pilot period are made using CREDENTIAL mobile app.

Acceptance criteria can be objective or subjective. In the second case, the user and her/his opinion plays a key role in the acceptance.

Expected Result	Acceptance Criteria
Citizen authenticates with CREDENTIAL Wallet IdP to SIAGE	- The Application Code of SIAGE hasn't been changed - A user can use any service of SIAGE using CREDENTIAL
Citizen uses CREDENTIAL mobile APP to authenticate	- Some authentications are performed by citizens using CREDENTIAL mobile APP; in other words, CNS smartcard is not the only authentication method used by citizens
Users finds "easy to use" and "appealing" the use of CREDENTIAL	- Customer satisfaction – the usage of CREDENTIAL mobile app would be easier and more attractive than the usage of a smartcard
Increased number of authentication from mobile devices	- The number of mobile authentication is relevant - the usage of CREDENTIAL mobile app would work as leverage to increase the number of authentication from mobile devices, which does not support smartcard authentication
Improved security and confidentiality	- Increased user perception of security and confidentiality - the use of CREDENTIAL cryptographic libraries and CREDENTIAL Wallet IdP would increase the security of the whole authentication solution
Cost reduction	- Cost reduction for Lombardy Region and LISPA: the use of CREDENTIAL IdP, which delivers no SMS – for example, would mitigate the cost of the entire authentication solution

Platform independence	- Access to SIAGE with CREDENTIAL Wallet IdP and CREDENTIAL App authentication is granted and available for any client (desktop) devices
Easiness to configure third-party IdP	- The configuration of third-party IdP is easy and could requires just adjustment or development – with reasonable effort - to the component called "IdP Adapter"
Easiness to configure another Lombardy Region SP	- Other Lombardy Region SP, previously integrated with IdPC, can be easily integrated to CREDENTIAL ecosystem and benefits of CREDENTIAL Wallet IdP authentication
Possibility to use the selective disclosure	- User can apply selective disclosure during online authentication, forwarding to SIAGE just a subset of the data retrieved by the Identity Provider

4.5 Key Performance Indicators Used to Evaluate the Pilot

Nine KPIs have been selected to evaluate the eGovernment pilot. Four KPIs are focused on technical aspects, while five KPIs are focused on the user perception of the pilot (like usability, etc.). The technical KPIs have a "success scenario" in which certain functionality has to be performed within a target, fixed time. The remaining KPIs have a "success scenario" in which a certain percentage of users selected to evaluate the pilot should declare satisfaction.

KPI EGOV-001	
Description	IdP Adapter – time needed to produce an authentication request to CREDENTIAL Wallet IdP
What will be measured	Time elapsed from SAML 2.0 authentication request - *received* by IdP Adapter - to authentication request *sent* to CREDENTIAL Wallet IdP
How it will be measured	Time elapsed written into IdP Adapter logs.
Reasoning	IdP Adapter translation process must be done in a reasonable time
Success Scenario	Time expected: <= 900 ms

KPI EGOV-002	
Description	IdP Adapter – time needed to produce an authentication response to Shibboleth SP
What will be measured	Time elapsed from authentication response - *received* by IdP Adapter - to authentication response *sent* to Shibboleth SP

How it will be measured	Time elapsed written into IdP Adapter logs. Time consumed by (optional) user selective disclosure will be not considered.
Reasoning	IdP Adapter translation process must be done in a reasonable time
Success Scenario	Time expected: <= 900 ms

KPI EGOV-003	
Description	Shibboleth SP – time needed to decrypt assertion values
What will be measured	Time needed to decrypt user identity data received from IdP
How it will be measured	Time elapsed written into Shibboleth SP logs - or similar
Reasoning	User data decryption must be done in a reasonable time
Success Scenario	Time expected: <= 900 ms

KPI EGOV-004	
Description	IdPC – time needed to perform proxy re-encryption of user identity data
What will be measured	Time needed to proxy re-encrypt user identity data, before SAML assertion is released
How it will be measured	Time elapsed logged into IdPC logs
Reasoning	Proxy re-encryption process must be done in a reasonable time
Success Scenario	Time expected: <= 1200 ms

KPI EGOV-005	
Description	Viability
What will be measured	User perception of CREDENTIAL compliancy with rules or laws; and the capacity to operate or be sustained in the future
How it will be measured	Specific questionnaire to focus groups – a typical five-level Likert scale should be used, for example: Strongly disagree; Disagree; Neither agree nor disagree; Agree; Strongly agree
Reasoning	This KPI is designed to investigate user perception of CREDENTIAL actual compliancy and sustainability in a mid-long term scenario
Success Scenario	>= 80% of people included in focus groups declares satisfaction

KPI EGOV-006	
Description	Usability

What will be measured	Quality attribute that assesses how easy user interface are to use, from a user point of view
How it will be measured	Specific questionnaire to focus groups – a typical five-level Likert scale should be used, for example: Strongly disagree; Disagree; Neither agree nor disagree; Agree; Strongly agree
Reasoning	CREDENTIAL applications must offer a reasonable trade-off between security and usability
Success Scenario	>= 80% of people included in focus groups declares satisfaction

KPI EGOV-007	
Description	Social acceptance
What will be measured	Limit to accept by a user the CREDENTIAL applications (App, web pages, …)
How it will be measured	Specific questionnaire to focus groups – a typical five-level Likert scale should be used, for example: Strongly disagree; Disagree; Neither agree nor disagree; Agree; Strongly agree
Reasoning	The criteria are designed to investigate the degree of acceptance of new tools and new process from a user point of view
Success Scenario	>= 80% of people included in focus groups declares satisfaction

KPI EGOV-008	
Description	Potential economic advantage
What will be measured	Cost reduction in future development of similar secure authentication systems
How it will be measured	Comparing actual cost of authentication systems in Lombardy Region with estimated cost of CREDENTIAL authentication system. For example, the usage of CREDENTIAL Wallet IdP instead of IdPC OTP/SMS mechanism results in a lower cost for Lombardy Region in term of € spent for every authentication (no more SMS needed)
Reasoning	The economic advantage should be a key factor in a potential, large-scale adoption of CREDENTIAL solution
Success Scenario	Estimated overall cost – for Lombardy Region and LISPA - of CREDENTIAL authentication system is lower than actual cost of Lombardy Region and LISPA authentication system

KPI EGOV-009	
Description	Perceived security

What will be measured	The added value of CREDENTIAL security from a user point of view
How it will be measured	Specific questionnaire to focus groups – a typical five-level Likert scale should be used, for example: Strongly disagree; Disagree; Neither agree nor disagree; Agree; Strongly agree
Reasoning	Often implemented security is not perceived as something “tangible” from users. One of the goal of the project should be to make “visible” and “tangible” the benefits of CREDENTIAL security
Success Scenario	>= 80% of people included in focus groups declares satisfaction

4.6 Pilot Execution

This section provides a general view of environment that will be used to deploy the LISPA eGovernment pilot, as well as details about the implementation (develop) strategy, the user-recruitment and a high-level time plan which summarize the activities.

The eGovernment pilot will be executed in a non-Production environment (namely “test environment”), mainly hosted by LISPA as described above. The test environment will be a replication of the LISPA’s latest production environment.

The implementations required in IdPC for the pilot will be developed within a branch started from the latest version of the production’s software, while the implementation of IdP Selector/Adapter will be specific for the pilot.

Two groups of users will be recruited: the first group will consist of LISPA personnel responsible for internal test execution, and the second group will consist of “key users”, created in a way to cover both technical and non-technical backgrounds.

The first group, consisting of at least 4 persons, will be responsible of the following activities:

- check if all eGovernment pilot requirements are covered
- check if all eGovernment pilot requirements are correctly implemented
- check if the eGovernment pilot is correctly usable from various types of devices, browser and operating systems

The second group, consisting of at least 16 voluntary persons ("key users"), will be recruited among these categories:

- SIAGE users
- IdPC users
- Lombardy Region personnel
- students (mainly from University)
- LISPA personnel not involved in CREDENTIAL project or other EU projects

It's important to underline that both user groups will use "fake" identities and "fake" data, in order to preserve privacy aspects.

The time plan for execution and verification will depend on interrelationship with other WP, with a special focus on cryptographic libraries and CREDENTIAL Wallet/IdP availability.

Most of the tasks, especially during the development phases, will be run in parallel, so we expect to manage overlapped activities.

The high-level time line is mainly divided into two major phases: Development and Integration / User test.

The development phase is structured as follow:

- In the Q2 2017 a first version of CREDENTIAL components will be made available from other partners. In this phase, it is important that the web services interfaces are defined and stable
- A first version of eGovernment pilot will be developed, starting from March 2017. This activity will last from 6 to 8 weeks. The CREDENTIAL components (software and documentation) available so far will be used
- A stable version of CREDENTIAL components should be available in Q3/early Q4 2017; eGovernment pilot will take benefit from this stable version and perform some adjustment and refinement
- The final result of development phase is the availability of a prototype of eGovernment pilot in the last quarter (Q4) of 2017.

Integration / User test is structured as follow:

- While the LISPA personnel start to test the eGovernment prototype (Q4 2017), the recruiting of the "key users" (already started in Q2-Q3) finish in the same period
- Every adjustment or refinement needed is develop and eGovernment pilot is update
- During the last months of 2017 and January 2018 "key users" are asked to test and evaluate eGovernment pilot
- Feedback from "key users" are gathered and discussed with partners (January/February 2018)
- Every adjustment or refinement needed is develop and eGovernment pilot is update (February-March 2018)

The eGovernment pilot will be available in its final version in March 2018.

4.7 Technologies

The eGovernment pilot will use the following technologies:

- **CNS – Carta Nazionale dei Servizi**: CNS is a national smartcard standard widely used in Italy. CNS is a ISO 7816 smartcard with a client authentication X509 certificate onboard to allow digital authentication over the Internet;
- **CSP – Crypto Service Provider:** CSP is a Windows software library that implements Microsoft CryptoAPI (CAPI). CSP is typically used to perform cryptographic operations, such as strong user

authentication and secure email. In LISPA eGovernment pilot, CSP is used to interface smartcard to establish an SSL client authentication, using smartcard private key. This usage needs certain browser, like Internet Explorer or Chrome;

- **PKCS #11** is a cryptographic token interface standard, which specifies an API, called Cryptoki (described here: http://docs.oasis-open.org/pkcs11/pkcs11-base/v2.40/os/pkcs11-base-v2.40-os.html). With this API, applications can address cryptographic devices as tokens and can perform cryptographic functions as implemented by these tokens. In eGovernment pilot, this technology is used to create a TLS/SSL client authentication, using smartcard private key, with certain browser, like Mozilla Firefox.
- **ISO 7816** is an international standard focused on contact id card, especially smartcard, managed and published by ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission); eGovernment pilot uses this standard because the usage of CNS requires an ISO7816 compliant smartcard reader;
- **eIDAS** interoperability Framework can be simplified as an extension and a modern generalization of the core of STORK Framework. The framework guarantees a cross-border authentication scenario;
- **SAML (Security Assertion Markup Language)** is an XML-based, open-standard data format for exchanging authentication and authorization data between parties, in particular, between an Identity Provider and a Service Provider. The eGovernment pilot widely uses this standard to assure secure user authentication between SP and IdP.

The eGovernment pilot also uses several components of CREDENTIAL Wallet platform, mainly for authentication and encryption purposes. These components are the *Authentication Service* and the *Data Management Service*. Other components that are likely used are the *Account Management* and the *Access Management Service*. These components are included in the whole architecture of CREDENTIAL Wallet, cf. also Figure 5.

The following lines provide some further details on the integration between the eGovernment pilot and the components mentioned above:

- Authentication Service is used when the CREDENTIAL Wallet platform is used as Identity Provider. The integration between these two components can be made in several ways and using several technologies, because the main task of LISPA component called IdP Adapter is to “translate” the native form of CREDENTIAL IdP authentication in a way which can be used by legacy Service Providers.
- Data Management Service is used both by Lombardy Region Identity Provider (IdPC) and Shibboleth SP to – respectively – crypt and decrypt user identity data. Please note that the usage of term “crypt” typically means “proxy re-encrypt”.
- Account Management and Access Management Service components are implicitly used because the citizen owns a CREDENTIAL account and manages this account.

The Technical Use Cases are focused on integration between LISPA component and CREDENTIAL platform, and can be summarized as follows:

- TUC to cover 7.A.1.2.1 and 7.A.1.2.2: IdPC access to CREDENTIAL platform to crypt user data - this functionality is used when Lombardian IdP needs to perform a proxy re-encryption of user data in order to transmit a complete identity assertion to a service provider ;
- TUC to cover 7.A.1.2.3: Shibboleth SP access to CREDENTIAL platform to decrypt data - this functionality is used when the Shibboleth SP receives the identity assertion with user data encrypted and needs to decrypt them in order to transmit plain identity data to a service provider ;
- TUC to cover 7.A.1.4.2: CREDENTIAL platform plays a role of and IdP and authenticates user - this functionality is used when a citizen decides to use the Identity Provider offered by CREDENTIAL platform ;
- TUC to cover 7.A.1.5.1: CREDENTIAL platform is used here as a secure store of additional user data; also, proxy re-encryption functionality is used - this functionality is used when an Identity Provider needs to access to additional user data, stored in the Wallet.

5. eHealth Pilot

The eHealth pilot is a web application to connect patients and physicians. In the pilot eHealth we concentrate on patients with Diabetes 2 as a disease and GPs and diabetes expert physicians. Data encryption falls far below most executive priority lists in health care organizations, like hospitals or private clinics. Clinics and GPs in Germany are not using encrypted software. Patients like to believe providers are eager to adopt encryption in healthcare, but the simple reality is that business executives and health providers aren't security experts. Encryption is vital to protect patient's data. However, everything boils down to usability. No matter how strong your security technology is, if it is too complex, or too difficult to use, it just won't work. Therefore, CREDENTIAL will offer a user-friendly possibility to exchange patient data in a secure way.

To evaluate the eHealth pilot, we will provide medical devices to measure blood sugar, blood pressure, weight and activities to 30 users. They will use it in their everyday life and provide health data to the database where GPs and health insurances will have access.

5.1 Scope

For the CREDENTIAL eHealth App end-to-end encryption based on CREDENTIAL Wallet and security serviced will allow patients, physicians, insurance companies and communities to access and share their medical data in a secure and privacy-preserving manner. Medical data, self-measured vital and movement data, and social community's data should be case-related integrated for an optimal patient's therapy. Possible use cases can be obesity prevention, diabetes monitoring or exercise therapy after femoral neck fracture or hip surgery. The goal of the use cases is to integrate data from different health care processes (e.g., medical treatment processes, accounting processes, community processes, etc.) in order to motivate the patient to a higher level of compliance. For example, a user of the CREDENTIAL eHealth App can provide her elf-collected data through a release mechanism to health care practitioners. The practitioner can compare the data with measurements of her diagnostic systems and can vouch for the success of prevention measurements and therapies. The patient can provide access to other data for insurance companies in order to confirm his compliance and to gain bonus points in suitable incentive programs. Communities should be involved in the CREDENTIAL eHealth App as part of compliance improvement and user motivation. For example, self-help groups for obesity prevention can be formed and members can share and compare their movement data in a competition scenario.

Stakeholders of the CREDENTIAL eHealth piloting scenario are doctors, medical personnel, health insurance companies and patients. A Personal Health Record will be available for all stakeholders to store patient data. By encrypting data using CREDENTIAL confidentiality protection means prior to storing data to the PHR, full end-to-end encryption can be achieved where only an authorized consumer of the medical data is able to decrypt the data. Providing such end-to-end encryption from data provisioning until data consumption is a prerequisite in some countries – e.g., Germany – for sharing health data via a cloud infrastructure. By this, CREDENTIAL is aimed as an enabler for establishing cost-effective and scalable cloud storage in healthcare.

Users for the piloting phase are patients, doctors, and medical personal. Depending on their individual care goal definitions, patients may be equipped with various Personal Health Devices. For the pilot digital scales, glucometers and pedometers will be considered.

5.2 Architecture Description

The eHealth pilot establishes a privacy-aware personal health record (PHR) that and contains all medical data provided by the patient or doctors within a defined treatment context. By using CREDENTIAL security services on data encryption, proxy re-encryption and authorization the PHR is equipped with means for end-to-end encrypted sharing of health data among patients and doctors. By this the PHR is fully under the control of the patient.

For the design of the solution, the following essentials have been formulated in order to gain high usability for all different groups of users:

- Users (patients and doctors) shall solely interact with business level services for sharing health information through a patient’s Personal Health Record (PHR). These services shall utilize accepted standards in the healthcare domain. All flows of control shall fully adapt to processes as defined by clinical guidelines and established in everyday practice.
- All security services shall be hidden to the user within an architectural layer that is only accessible through business functions. By this all security configuration is derived from business processes and settings. For instance, if the user names a doctor as his family doctor, this doctor is granted full access permissions to the user’s health record. All technical details and messages for setting respective permissions are fully encapsulated through internal interactions among eHealth business services and CREDENTIAL security services.
- Health Information Exchange Services shall not implement their own security services but fully rely on CREDENTIAL services for user authentication, user authorization and data confidentiality protection. Utilization of CREDENTIAL security services is automatically triggered by intercepting any access attempt to protected resources.

Figure 12 sketches the overall setup of the CREDENTIAL eHealth pilot consisting of two layers and two tiers. It shall be recognized that for this pilot, the business services and security services which were considered as tiers in the CREDENTIAL generic architecture have been layered for fully decoupling the respective services. This is a prerequisite to reach the pilot’s goal of fully integrating CREDENTIAL with existing healthcare-IT standards.

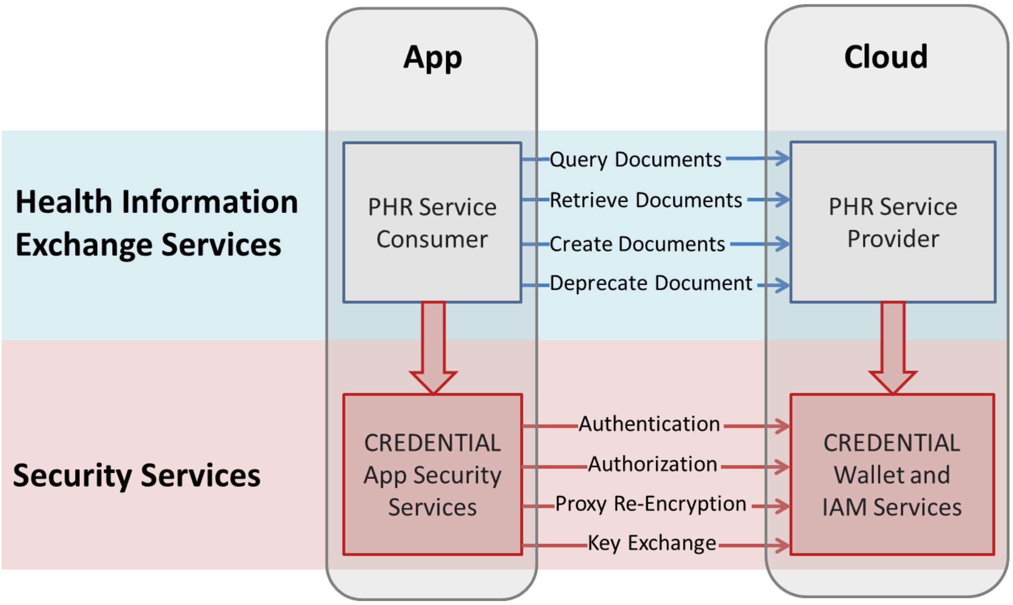


Figure 12: Layers and Tiers for the eHealth Pilot

The consumer side as the front-end tier of the solution is provided through a mobile App. On the Health Information Exchange layer, services are located that

- translate user interactions into standard messages and data formats for interacting with the backend PHR services
- integrate CREDENTIAL security services and wallet services into the flow of control in order to safeguard access to the PHR and to provide full end-to-end encryption

The consumer side CREDENTIAL App Security Services package the services usually deployed within the CREDENTIAL generic App into kind of a library that is fully capsuled behind the GUI of the CREDENTIAL eHealth App. Private keys are solely managed and used through services deployed within the CREDENTIAL App Security Services. All requests to cloud-based CREDENTIAL services (e.g., for sharing PHR-Keys through the Wallet) are routed through this library in order to clearly separate business functionality from security mechanisms.

The CREDENTIAL eHealth App will be operated in two defined environments:

- The patient environment is made up from a smartphone owned by the patient, the CREDENTIAL eHealth App running on the smartphone, and various personal health devices for collecting medical data and monitoring the patient. The CREDENTIAL eHealth App assists the patient by accessing his PHR and capsules authentication and authorization through its GUI. The main purpose of the CREDENTIAL Patent App is to enable the patient to upload monitoring data, to visualize and analyse his own data, and to interact with his doctors.
- The doctor environment is made up from a table used by the doctor and the CREDENTIAL Doctor App running on the tablet. The CREDENTIAL Doctor App allows a doctor to manage the

patients involved in the CREDENTIAL eHealth pilot, to visualize and analyse the progress of a patient's therapy, and to share data with his patients.

The backend tier of the solution consists of standard PHR services and the common CREDENTIAL cloud-based services (Wallet and IAM). The PHR services will be provided through an existing open source health record solution that is configured to match the CREDENTIAL requirements. By this the CREDENTIAL eHealth pilot will utilize an almost off-the-shelf backend with defined interfaces to the CREDENTIAL Wallet.

5.3 Selected Use Cases

Diabetes uses cases were chosen because rates of type 2 diabetes have increased markedly since 1960 in parallel with obesity [2]. As of 2013 there were approximately 368 million people diagnosed with the disease compared to around 30 million in 1985 [3]. Typically, it begins in middle or older age [4] although rates of type 2 diabetes are increasing in young people [5]. Type 2 diabetes is associated with a ten-year-shorter life expectancy [6].

The outcome of the use cases will be to move the data and not the patient.

5.3.1 Health Data Collection and Sharing

The use case "**Care Planning and Progress Tracking**" highlights the collection and sharing of diabetes-related data. It functions basically as the starting point for patients to participate with the CREDENTIAL Wallet. It covers the aspects that a new Patient Health Record (PHR) has to be created for a patient, that the eHealth app needs to be installed and configured for the PHR and that the medical devices needs to be connected to the eHealth app. All following use cases build up on the success of this use case.

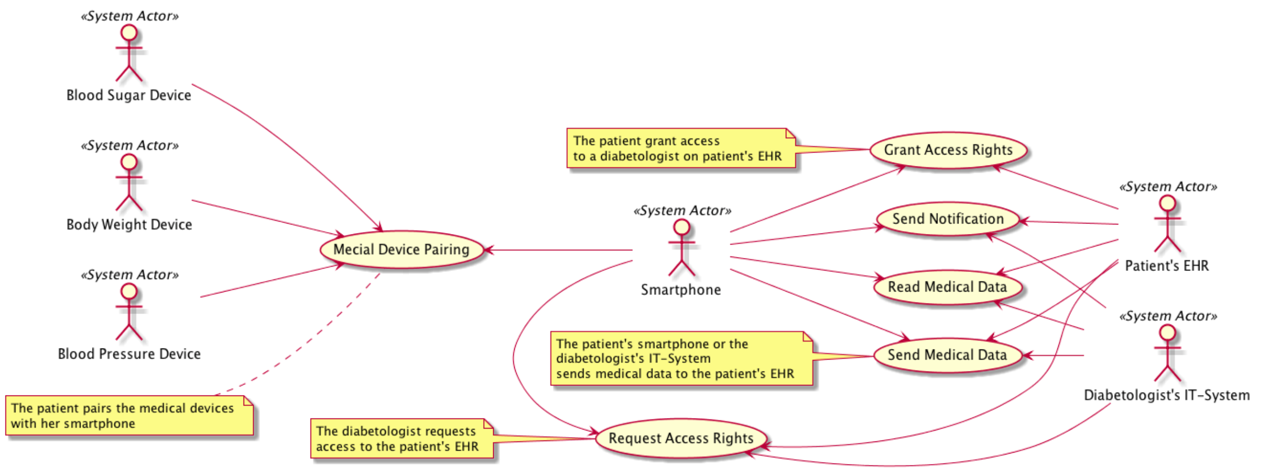

«System Actor»
Blood Sugar Device
«System Actor»
Body Weight Device
«System Actor»
Blood Pressure Device
Mecial Device Pairing
The patient pairs the medical devices with her smartphone
The patient grant access to a diabetologist on patient's EHR
Grant Access Rights
«System Actor»
Smartphone
Send Notification
Read Medical Data
Send Medical Data
The patient's smartphone or the diabetologist's IT-System sends medical data to the patient's EHR
The diabetologist requests access to the patient's EHR
Request Access Rights
«System Actor»
Patient's EHR
«System Actor»
Diabetologist's IT-System

Figure 13: Care Planning and Progress Tracking Use Case Definition

In the "**Nutrition and Activity**" use case, the patient collects what she does (ex: steps) and what she eats. The data is provided by the patient and shared with her doctor. There are multiple ways to get to such data:

- The patient enters the data manually.
- The app could directly communicate with relevant activity sensors (ex: step counter).
- As there are many other good apps, data might be extracted from those apps.
- Mobile platforms might offer a health database, which already collects such data.

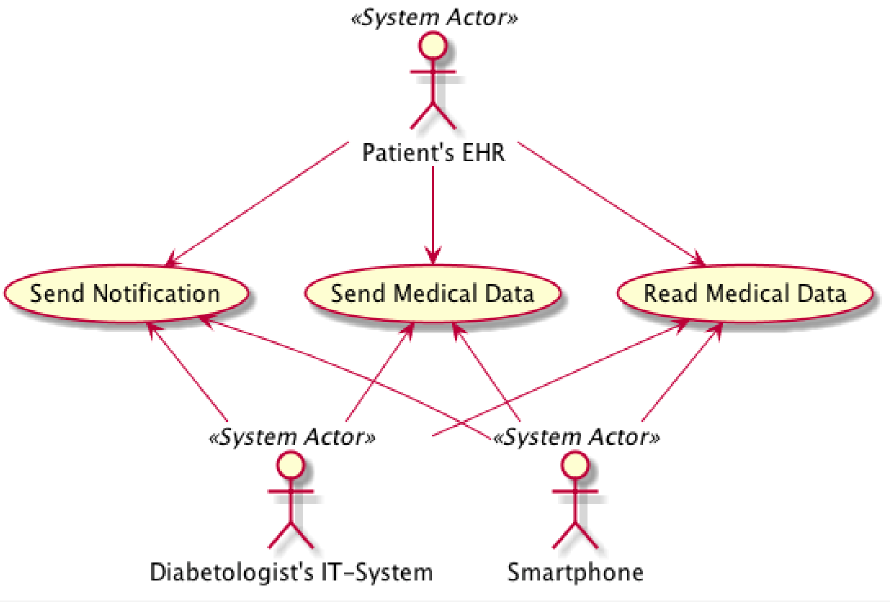


Figure 14: Nutrition and Activity Use Case Definition

5.3.2 Notification

The use case "**Therapy Monitoring and Screening for Complications**" focuses on the doctor's review process. The goal is to support the doctor in his decision making by the notification mechanism of the CREDENTIAL Wallet. Thus, we hope that a doctor can make more informative decision for a particular medical treatment of his patients. Together with doctors we find out which information in today's diabetes treatment may help the doctor. Here, the doctor is notified when the patient's data exceeds a threshold for a defined timespan. For example this might be the case when the patient's blood pressure is too high for more than 3 days. In addition, notifications when a patient does not provide data over a particular timespan might be helpful.

The diabetologist is responsible for monitoring and documenting the progress of therapy. This includes coordinating visits to other specialists, in particular for regular screening for complications and co-morbidities. Not only in the case of a managed care program but even based on regular health legislation, the diabetologist is obliged to provide information about the patient in case of a referral to a specialist or a hospital for diabetes-related diagnostics or interventions. The monitoring will be done manually. Thresholds will be registered manually as well.

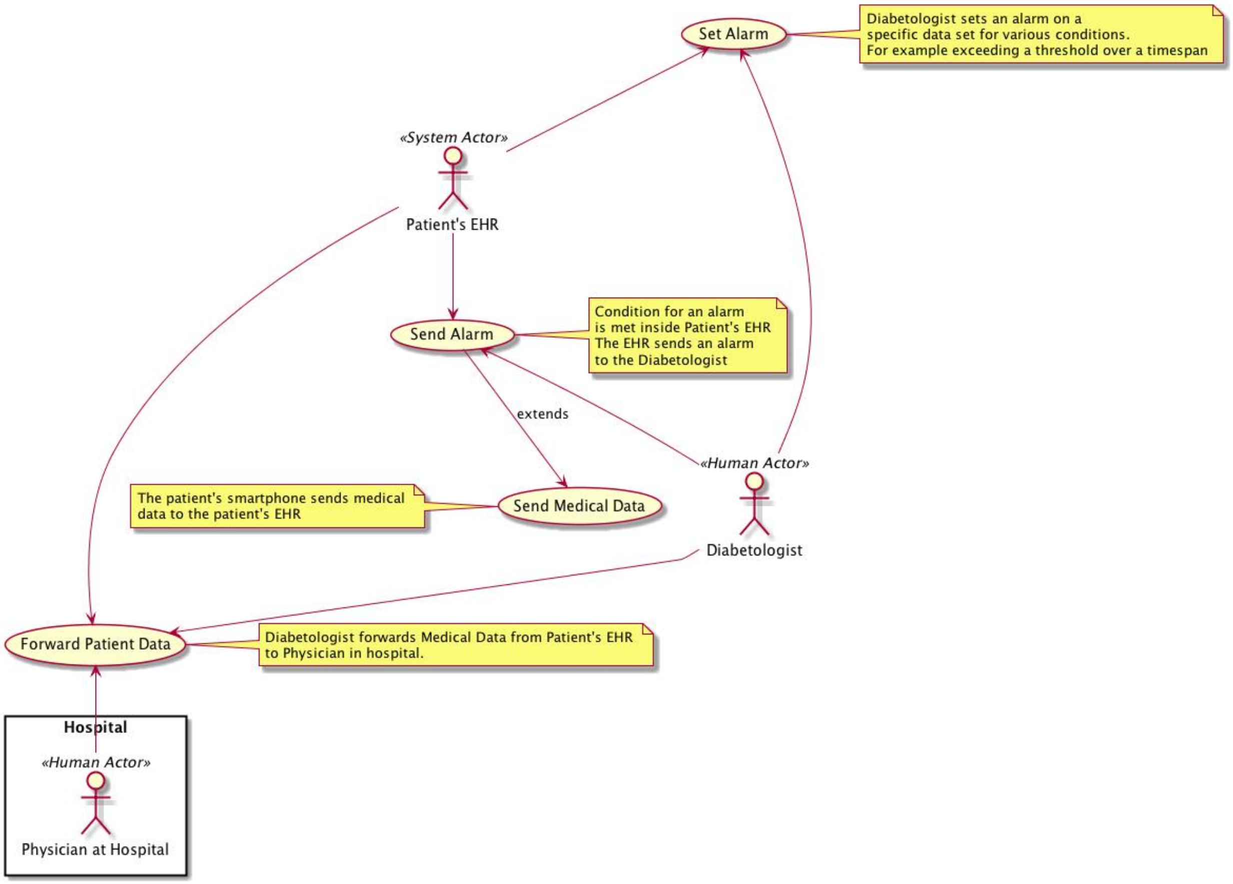

Set Alarm
Diabetologist sets an alarm on a specific data set for various conditions. For example exceeding a threshold over a timespan
«System Actor»
Patient's EHR
Send Alarm
Condition for an alarm is met inside Patient's EHR The EHR sends an alarm to the Diabetologist
extends
«Human Actor»
Diabetologist
The patient's smartphone sends medical data to the patient's EHR
Send Medical Data
Forward Patient Data
Diabetologist forwards Medical Data from Patient's EHR to Physician in hospital.
Hospital
«Human Actor»
Physician at Hospital

Figure 15: Therapy Monitoring and Screening for Complications Use Case Definition

5.3.3 Medication Plan

From stage 2 on diabetes patients are treated with medication (initially Metformin which may later be complemented by other substances). Given that many of the factors that favour the severity of diabetes type-2 (e.g., obesity, smoking, low activity) may also advance other diseases, diabetes type-2 patients (some patients have a combination of diabetes type-1 and type-2) often suffer from additional problems which may require medication. Therefore, it can be assumed that many diabetes patients will regularly take at least three different pharmaceutical products.

German legislation obliges doctors and pharmacists to provide such patients with a (paper-based) medication plan in order to support the patient in taking his medication properly but even to enable checks for interdependencies, contra-indications and adverse reactions.

Medications for Hyperglycemia in Type 2 Diabetes:

- Metformin: Preferred initial therapy (if tolerated and not contraindicated) when lifestyle changes alone have not achieved or maintained glycemic goals
- Consider insulin therapy with or without other agents. At outset in newly diagnosed patients with markedly symptomatic and/or elevated blood glucose levels or A1C

- Add 2nd oral agent, GLP-1 receptor agonist, or insulin. If noninsulin monotherapy at maximal tolerated dose does not achieve or maintain A1C target over 3 months

Doctors review this plan when adding another prescription to check for contra indication.

5.4 Details on the Selected Use Cases

This section covers more technical details on the selected use cases of the eHealth pilot. A detailed specification can be found in the Appendix B.

5.4.1 Managing Permissions

The CREDENTIAL eHealth pilot uses a combination of Role-Based Access Control (RBAC) and Discretionary Access Control (DAC):

- The patient may assign a role to a user. The user is granted all permissions linked to that role. Initially only the following roles will be used:
  - Family Doctor, Care Manager: these roles are granted full access to all documents within the patient's PHR. In the CREDENTIAL diabetes use case the diabetologist takes the role "Care Manager". A patient may link multiple doctors with each role. For resident physicians, these roles may be assigned to the practice rather than to the doctor as a person.
  - Registered Doctor: doctors and organizations listed with the patient's address book may upload documents to the patient's PHR. They may as well update or deprecate documents for which they are named the author.
  - NULL: For taking all permissions from a doctor, this doctor is assigned a NULL-role. Doctors with this role may not access a patient's PHR and will not even be given information on the existence of a PHR for the patient. Initially all doctors registered to CREDENTIAL are assigned this role with respect to all PHRs managed through CREDENTIAL (fail-safe default).
- The patient may grant discretionary and temporal access to a user for a single document. The user may only download this document. The user shall not access any other documents (unless respective permission is given) and shall not be able to see the table of content of the patient's PHR.

Internally all permissions (RBAC and DAC) are mapped onto a policy-based access control paradigm. For this the CREDENTIAL Patient App translates the granted RBAC- and DAC-style permissions to XACML policies, which are then registered with the CREDENTIAL Policy Administration Point within the CREDENTIAL Authorization Service.

For defining, listing and deleting permissions, solely generic CREDENTIAL services will be used. Nevertheless, it is the responsibility of the CREDENTIAL Patient App to map the XACML policies as managed by CREDENTIAL to RBAC and DAC rules for maintaining the patient's address book.

5.4.2 Selection Justification

- The “**Care Planning and Progress Tracking”** use cases were selected because the control of body weight and of blood glucose concentrations depends on the coordination of the function of several organs and tissues, in particular the liver, muscle and fat. These organs and tissues have major roles in the use and storage of nutrients in the form of glycogen or triglycerides and in the release of glucose or free fatty acids into the blood, in periods of metabolic needs. These mechanisms are tightly regulated by hormonal and nervous signals, which are generated by specialized cells that detect variations in blood glucose or lipid concentrations. The hormones insulin and glucagon not only regulate glycemic levels through their action on these organs and the sympathetic and parasympathetic branches of the autonomic nervous system, which are activated by glucose or lipid sensors, but also modulate pancreatic hormone secretion and liver, muscle and fat glucose and lipid metabolism.
- The use case “**Therapy Monitoring and Screening for Complications**” was chosen because obesity and type 2 diabetes mellitus (T2DM) are the leading worldwide risk factors for mortality. For the best outcome, continuous glucose monitors (CGM) use should be on a regular basis to improve glucose control and reduce hypoglycemia [7] in both type 1 and type 2 diabetes patients.

  The inextricably interlinked pathological progression from excessive weight gain, obesity, and hyperglycemia to T2DM, usually commencing from obesity, typically originates from overconsumption of sugar and high-fat diets. Although most patients require medications, T2DM is manageable or even preventable with consumption of low-calorie diet and maintaining body weight. Medicines like insulin, metformin, and thiazolidinedione that improve glycemic control; however, these are associated with weight gain, high blood pressure, and dyslipidaemia [8].
- The use cases “**Oral Medication**” and “**Insulin Therapy**” were selected because it is very important to adjust the medication according to the health data of the patient. As those data will be transmitted electronically, the feedback for the patient should be within short time after the decline of the health data. The medication plan will be updated to include the insulin that is prescribed. The frequency of blood sugar measurements is adapted, so that the patient will need to measure her blood sugar 30 minutes before every meal, 2 hours after every meal and 1 hour before going to bed.

5.4.3 CREDENTIAL Service Functional Model

The Figure 16 sketches the Service Functional Model (SFM) of the CREDENTIAL health information exchange services that reflect the logical services to be implemented for the eHealth pilot. Business services based on standard eHealth data sharing protocols are labeled in blue while security services on top of the CREDENTIAL Wallet are signaled by red arrows.

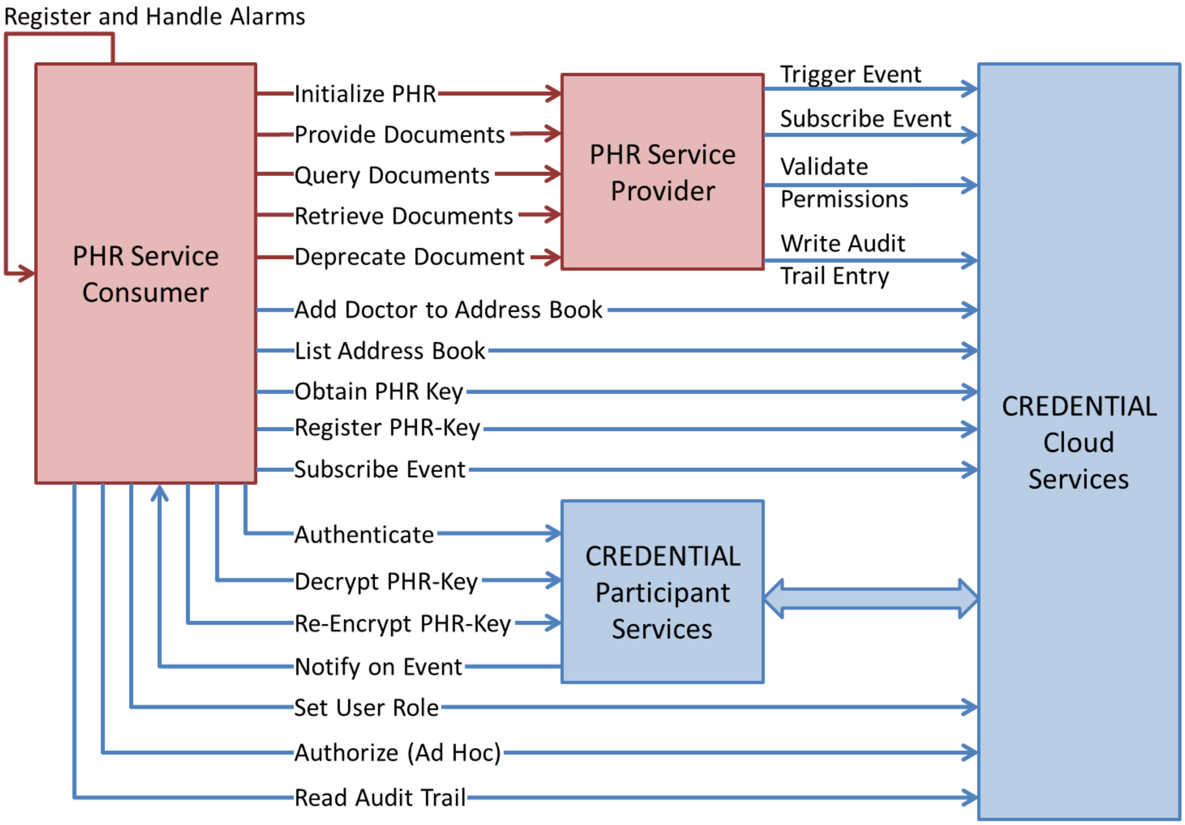


Figure 16: Services as Implemented by the CREDENTIAL eHealth Pilot

The table below gives a short overview on these services.

Service	Description
Initialize PHR	Set up a PHR instance within a CREDENTIAL account
Provide Documents	Upload a set of medical documents to a patient’s Personal Health Record (PHR)
Query Documents	Discover documents within a patient’s PHR that match with given query parameters (e.g., type of document)
Retrieve Documents	Download a set of identified documents from a patient’s PHR
Deprecate Documents	Mark a set of documents as invalid or outdated
Authenticate	Authenticate a user
Set User Role	Assign a role (e.g., Family Doctor) to an identified user and grant all permissions linked with this role
Authorize (Ad Hoc)	Grant access rights to an identified user for accessing a single document
Obtain PHR Key	Make the PHR-Key available to an authorized user
Register PHR-Key	Store a PHR-Key as a protected document to the CREDENTIAL Wallet.
Decrypt PHR-Key	Decrypt a PHR-Key using CREDENTIAL crypto mechanisms.
Re-Encrypt PHR-Key	Make a PHR-Key available to an authorized user be re-encrypting it for being decrypted with that user’s private key.
Validate Permissions	Assess an access request with respect to the policies as defined by the patient

Write Audit Trail Entry	Write an audit trail entry to the CREDENTIAL Wallet
Read Audit Trail	Read all audit trail entries written for an identified patient's PHR.
Add Doctor to Address Book	White/yellow pages search for a doctor in the CREDENTIAL participant directory and inclusion of this doctor to the patient's address book. Note that there is no Address Book feature in the CREDENTIAL Wallet itself. Thus, this service will be created through the access control management services and the address book will be stored as an own policy.
List Address Book	Read a list of all CREDENTIAL participants who had been registered by the patient as his doctors. Note that there is no Address Book feature in the CREDENTIAL Wallet itself. Thus, this service will be created through the access control management services and the address book will be stored as an own policy.
Subscribe Event	Subscribe to an event (e.g., new document added to an account). The PHR service provider maps the respective IHE transaction to the wallet API.
Trigger Event	Forward a PHR-issued event to the CREDENTIAL Wallet which takes responsibility for sending out respective notifications.
Notify on Event	Notify a user about an event he subscribed to.
Register Alerts Handle Alerts	Doctors may register alerts with their CREDENTIAL Doctor App that fire when a defined condition is met (e.g., a lab value is above a certain threshold).

Further services such as the local registration of alert conditions or the management of address books have been defined as part of the CREDENTIAL eHealth business use cases. As these services do not interact with CREDENTIAL and/or the PHR provider services, their further elaboration is left to the implementation of the CREDENTIAL eHealth App.

The table below summarizes which CREDENTIAL services and IHE actors are utilized for providing the logical services functionalities. This table again manifests the strict decoupling of IHE based business and CREDENTIAL based security services as each logical service clearly maps to either of these architectural layers (event subscription is no exception, as the business service just acts as an intermediary to map the IHE standard interface to the proprietary CREDENTIAL service interface).

Service	CREDENTIAL Services Used	IHE Actors/Transactions Used
Initialize PHR		- Identity feed to the Document Registry - Provide and register document set to the Document Repository
Provide Documents		- Provide and register document set to the Document Repository
Query Documents		- Stored query to the Document Registry
Retrieve Documents		- Retrieve document set from the Document Repository

Deprecate Documents		- Metadata update to the Document Registry
Authenticate	- Identity Provider for user authentication - Attribute Service for obtaining user identity attributes	
Set User Role	- Policy administrator for registering permissions - Key management for creating and registering re-encryption key	
Authorize (Ad Hoc)	- Policy Administration Point for registering permissions - Key management for creating and registering re-encryption key	
Obtain PHR Key	- Wallet for retrieving PHR-Key - Proxy Re-Encryption - Crypto services for decrypting PHR-Key	
Validate Permissions	- Policy Decision Point	
Register PHR Key	- Wallet for storing PHR-Key	
Write Audit Trail Entry	- Audit Service for storing audit trail entry	
Read Audit Trail	- Audit Service for providing log entries - Decryption service	
Add Doctor to Address Book	- Participant directory for white and yellow pages search	
List Address Book	- Policy Administration Point for querying authorized users	
Subscribe Event	- Event Manager for registration of event subscriptions	- XDS Façade for triggering events related to IHE XDS transactions
Trigger Event	- Event Manager for triggering a notification	- XDS Registry upon document metadata change
Notify on Event	- Event Manager for sending out notifications	
Decrypt PHR-Key	- Crypto services for decryption with a user's private key	

Re-Encrypt PHR-Key	- Proxy Re-Encryption	
Register Alerts	- Notification Service for subscribing on events	
Handle Alerts		- XDS Registry and Repository for retrieving relevant data

5.4.4 Integration of Data from Personal Health Devices

For the CREDENTIAL eHealth pilot, patients will be equipped with Personal Health Devices:

- A glucometer allows the patient to regularly measure his blood sugar at home. It is assumed that each participating patient will take 3-6 measurements per day. Measured data is stored with the device and forwarded via Bluetooth to a paired smartphone as soon as this is within reach.
- Patients are advised to measure their body weight at least once a day using a digital scale. The scale used for the pilot operation stores its data to a vendor cloud from which it can be read via the patient’s smartphone.
- Depending on the patient’s co-morbidities some patients will additionally be given a pedometer or a fitness tracking device for counting their daily steps count. Data from these devices is sent to the patient’s smartphone via Bluetooth as soon as the phone is paired and in reach.

The CREDENTIAL eHealth Pilot’s architecture follows the Continua Health Alliance’s reference model for integrating personal health devices with a personal health record infrastructure. Based on this approach any device data is first collected by an Aggregation Manager which gathers data from different devices and integrates them into structured documents. These documents are then transmitted to the personal health record through a personal gateway on the smartphone which takes responsibility for connection establishment, connection security and data transmission over local and wide area networks.

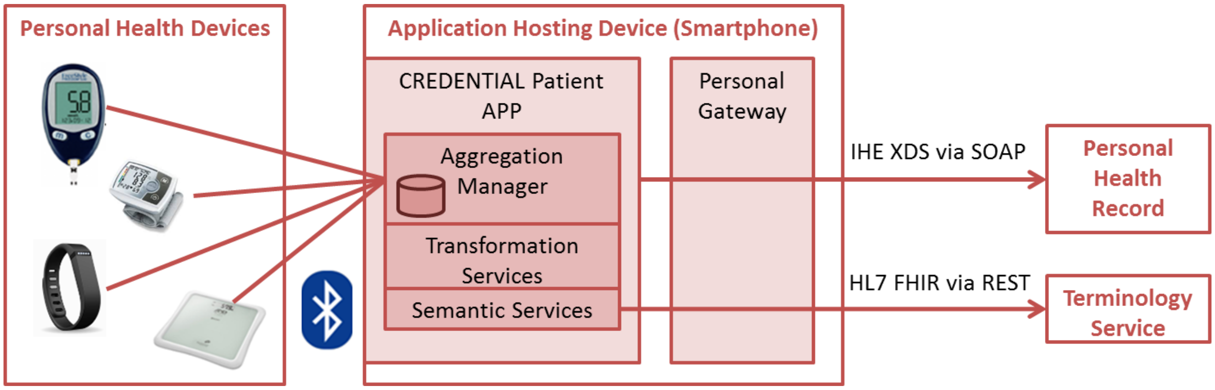


Figure 17: Aggregation and Transformation of Device Data

The Aggregation Manager logical components uses local transformation services for transforming raw monitoring data into structured HL7 FHIR resources which again can be bundled as documents. For fostering semantic interoperability with the data consuming services, all monitoring data is coded using standard terminologies. The respective mappings are performed by Semantic Services within the Patient App which loads the target code system information from a centrally managed Terminology Service.

5.5 Expected Results

The goal of the eHealth pilot is to show the integration of existing eHealth applications and standards with the security mechanisms provided by the CREDENTIAL Wallet.

Expected Result	Acceptance Criteria
Patients can share self-collected medical data with their doctors.	The PHR Key is encrypted in the CREDENTIAL Wallet. The authorization mechanism of the CREDENTIAL Wallet enables the declaration of specific access rules for doctors on the PHR Key. The CREDENTIAL Wallet is able to re-encrypt the PHR Key for a patient's PHR to a doctor.
Doctors are informed about the status of patient's medical data.	The notification service in the CREDENTIAL Wallet sends a notification to doctors when a patient provides new data or over a timespan no data was added by the patient. The doctor's IT-System uses the provided information to analyse the newly added data for exceeding certain thresholds.
Authentication of Patients and Doctors is performed with the CREDENTIAL IdP.	Authentication requests performed by the eHealth App for patients and doctors are delegated to the CREDENTIAL IdP.
Authorization decisions to PHR Services is performed with CREDENTIAL Wallet security services.	The Authorization Services of the CREDENTIAL Wallet are able to process and evaluate policies which reference external resources from the PHR services. Authorization requests from the PHR services are delegated to the CREDENTIAL security services.
Hospitals and Doctors will have health data available which can be used for statistics and epidemiology. Mapping by inserting data in a GIS will be possible.	CREDENTIAL will send data automatically to health insurance companies or any other authorized organisation.

5.6 Key Performance Indicators Used to Evaluate the Pilot

The KPIs build the basis for the technical evaluation as well as the user feedback evaluation for the eHealth pilot. Challenges are the technical integration as well as the helpful feature enhancements by eHealth applications in the treatment of diabetes patients. The following list of KPIs will be measured through the two phases of the eHealth pilot.

Description	Average decryption time of medication plan on patient eHealth App
What will be measured	The decryption time of each decryption process of a medication plan. The size of the medication plan in Kbyte.

How it will be measured	After the eHealth App receives the medication plan the decryption process is started and the elapsed time is written in a log entry. We analyse the log entries afterwards or by a weblog support tool like kibana [9].
Reasoning	A fast decryption leads to more user satisfaction.
Success Scenario	Average time: <= 1000 ms

KPI EHLTH-002	
Description	Participants contributing glycosometer data
What will be measured	The number of participants who contributes at least one glycosometer data per day
How it will be measured	A log entry is written if a glycosometer data entry is stored in the CREDENTIAL Wallet
Reasoning	The quality of the research questions can only be guaranteed if the participation is high enough.
Success Scenario	>= 51% of participants contribute at least one glycosometer data entry per day

KPI EHLTH-003	
Description	Participants contributing body weight data
What will be measured	The number of participants who contributes at least one body weight data per day
How it will be measured	A log entry is written if a body weight entry is stored in the CREDENTIAL Wallet
Reasoning	The quality of the research questions can only be guaranteed if the participation is high enough.
Success Scenario	>= 51% of participants contribute at least one body weight data entry per day

KPI EHLTH-004	
Description	Participants contributing activity data
What will be measured	The number of participants who contributes at least one activity data per day
How it will be measured	A log entry is written if a activity entry is stored in the CREDENTIAL Wallet
Reasoning	The quality of the research questions can only be guaranteed if the participation is high enough.
Success Scenario	>= 51% of participants contribute at least one activity data entry per day

KPI EHLTH-005	
Description	Re-Encryption time of PHR Key
What will be measured	The average time of a re-encryption procedure of a PHR Key within the CREDENTIAL Wallet.

How it will be measured	The CREDENTIAL Wallet writes log entries whenever an entry is re-encrypted for another participant.
Reasoning	The re-encryption time has influence on the request response time.
Success Scenario	Average time: <=500 ms

KPI EHLTH-006	
Description	Perception of enhanced privacy
What will be measured	The indicator of the participants' perception of the enhances privacy through the CREDENTIAL Wallet
How it will be measured	A questionnaire to participants. A typical five-level Likert scale should be used, for example: Strongly disagree; Disagree; Neither agree nor disagree; Agree; Strongly agree
Reasoning	Investigate the perception of participant's awareness of enhanced privacy through CREDENTIAL Wallet.
Success Scenario	>= 80% of participants declare satisfaction.

KPI EHLTH-007	
Description	Willingness to share data with participants
What will be measured	The indicator of the participants' willingness to share their medical data with other stakeholders, for example insurance companies
How it will be measured	A questionnaire to participants. A typical five-level Likert scale should be used, for example: Strongly disagree; Disagree; Neither agree nor disagree; Agree; Strongly agree
Reasoning	The CREDENTIAL mechanisms allows privacy and security aware sharing of data without exposing the content to the service provider.
Success Scenario	>= 51% of participants declare willingness; or reasons why participants does not want to share data with other stakeholders are clear

5.7 Pilot Execution

The eHealth pilot will be executed in two phases. The phases are explained in the following sections.

5.7.1 First Phase Approach

The first phase approach is the theoretical part. The consortium agreed to collect data from test users. We are executing the pilot with real data from real people. The big challenge for the eHealth pilot is to integrate existing technologies and standards with the CREDENTIAL Wallet components and develop tailored solutions for the elaborated use cases. Therefore, the first phase is prefixed by a development phase.

Since the eHealth pilot uses a highly customised user interface the development phase starts by implementing the user interface for the patient and the doctors. They are mainly developed by the experience of the partners Fraunhofer FOKUS and Klughammer. Doctors with experience in the medical treatment of diabetes patients will be integrated in the design process in this phase.

While the generic UI is developed in parallel to the eHealth pilot apps, it will be analysed which generic UI components can be reused in the eHealth pilot by not breaking the usability in the certain scenarios.

The goal of the first phase is to find bugs, get feedback from the participants and build up a lessons learned catalogue which will be the input for the second phase. The lessons learned catalogue is based on the KPI defined in Section 5.6 and steering forward to give answers to the research questions.

5.7.2 Second Phase Approach

The second phase approach is the practical part. The pilot will be executed with 30 participants acting as patients. We will recruit them through contacts to hospitals, doctors and students made by Klughammer. Each participant is equipped with an Android Smartphone, the CREDENTIAL eHealth app, a glycosometer, a body weight, and an activity tracker. Together with doctors we will establish a common exercise plan for a usual diabetes patient. The participants are instructed to use the eHealth app according to the elaborated exercise plan. Over a time span of three months the participants use the eHealth app under the guidance of their doctors. After three months, we collect feedback from the participants and analyse how the eHealth app well performed according to the KPIs. During the next two months, we correct bugs and add features wanted by the participants. Next the participants are again instructed to execute their exercise plans over a timespan of two months. At the end, we collect the feedback and evaluate the research questions and the success of the eHealth pilot.

During the pilot execution, various research questions will be target of scientific research. The main focus is put on the privacy aspects and self-determination of a patient’s medical data. The patient entrusts the cloud provider with responsible handling of the medical data. Despite the privacy challenges for the CREDENTIAL Wallet provider the proposed solution offers great opportunities for patients and medical treatment. For example, health insurance companies are highly interested in efficient and successful measurements but lack the possibility to verify and control certain results. The research will analyse if patients are willing to share their medical data.

- Do patients handle their personal medical data more carefully, if they have more control about data flows and addressees of their medical data?
- Which types of incentive concepts do motivate users to share their personal data with payers in the health domain?
- How fine-granular should access policies be designed, such that individual user needs are maintained and users do not hesitate to use the features of a Cloud Identity Wallet due to complicated usage?
- Is a safe cloud storage solution a decision criterion for users to use health apps and platforms from this specific provider?

- Can the integration of therapists increase the quality of "Quantified Self" data?
- Is there a positive health impact in the medical treatment relationship between patient and doctor by integrating a notification mechanism about the patient's medical data?

These research questions will be analysed by questioners towards the participating patients and doctors. The questions will be defined before the first pilot phase.

The user recruitment plan looks like the following:

Recruitment		
User I	Patients	Users will be recruited by KGH e.g., students and other voluntaries
User II	Physicians	KGH will inform GPs and diabetologists about the testing an innovative and secure eHealth platform for patients with diabetes II.
User III	Insurance Company	KGH will inform insurance companies about testing an innovative and secure eHealth platform for patients with diabetes II.

The pilot execution plan looks as follows:

Execution			
User I	Patients	Smart Phone	One-time registration and log-in at the physician's office. Devices used by the patients at home and the smart phone used will be synchronized. Patient will use her/his smart phone to collect data at home from her/his Bluetooth devices - scale - blood pressure - blood sugar Data will be transferred automatically to her/his actually treating physician. Patient will not fill out any form. Patient will give different allowance to physicians and insurance how the transferred data may be used. - for treatment - for statistical use (physician internal only) - for statistical use (insurance internal

			only) - for national diabetes register
User II	Actually treating physician	Tablet/Smart Phone	Will be given the allowance from patient to have access to the data transmitted by the devices.
User III	Insurance Company	Tablet/Smart Phone	Will be given the allowance from patient to have access to the data transmitted by the devices.
User IV	Physicians This can be a physician whom the patient will consult because of emergency situation, when the actually treating physician is not available.	Tablet/Smart Phone	Patient will authorize the physician to have access to data defined by the patient.

5.8 Technologies

This section gives a short overview on the eHealth standards used in the CREDENTIAL eHealth pilot. Further background about why just these standards have been chosen can be found in the CREDENTIAL deliverable "D4.1: Assessment report on cryptographic technologies, protocols and mechanisms".

5.8.1 IHE XDS: Cross-Enterprise Document Sharing

The IHE profile "Cross-Enterprise Document Sharing (XDS)" defines interfaces, metadata and data flow protocols for sharing medical documents [10]. IHE XDS builds upon the OASIS ebXML standard and provides a specific configuration and deployment of this standard's components to reflect typical use case within regional healthcare networks.

IHE XDS is the de facto healthcare-IT standard for record based infrastructures. It is widely used worldwide and has broad support from healthcare-IT-vendors. In Europe, many health information exchange networks build upon IHE XDS and most of the public tenders for such networks request for IHE XDS as the base standard.

The CREDENTIAL eHealth pilot will use an IHE XDS compliant backend for managing medical data. The figure below sketches how standard IHE actors and transactions will be integrated with the overall solution. The XDS Document Registry and Document Repository actors will be capsuled by an XDS Façade that mediates the integration of the CREDENTIAL IAM services and wallet.

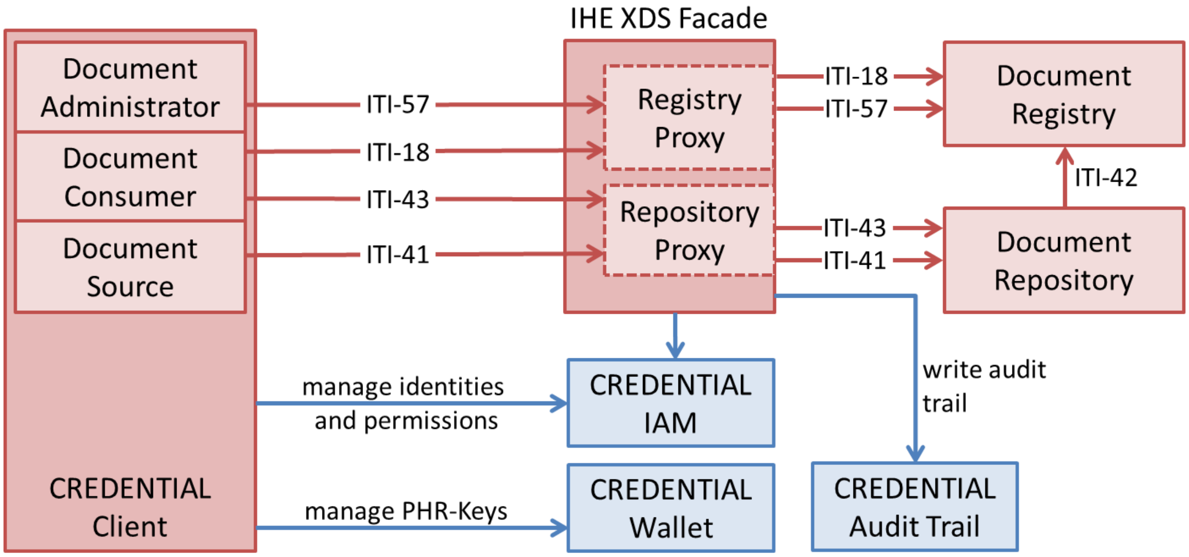


Figure 18: IHE Actors and Transactions Used for Implementing the CREDENTIAL eHealth Pilot

5.8.2 IHE ATNA: Audit Trail and Node Authentication

The IHE ATNA integration profile introduces the notion secure nodes and secure applications. A secure application is accessible only from authenticated nodes and logs all access attempts to protected resources with a secure audit trail. A secure node reflects a processing environment that only operates secure applications.

IHE ATNA builds upon TLS mutual node authentication for safeguarding access to secure applications and the syslog protocol (RFC5424-5426) together with DICOM - 2011 PS 3.15 (Part 15), Annex A.5 (ISO 12052) for writing audit trail messages.

For CREDENTIAL only the concept of a secure application will be used while the technical means for implementing the provided health information exchange services as secure applications will build upon “plain” TLSv1.2 for node authentication and CREDENTIAL Wallet services for writing the audit trail. The use of the CREDENTIAL Wallet imposes the advantage that no additional means are required for safeguarding the audit trail as CREDENTIAL easily allows for saving audit trail entries in a manner that only allows for the patient to read this data.

5.8.3 IHE DSUB: Document Metadata Subscription

IHE DSUB builds upon the OASIS Web Services Notification family of standards and defines actors and transactions (logical components and services/messages) for subscribing to events, publishing events and transmitting notifications on events.

The CREDENTIAL eHealth pilot will fully rely on CREDENTIAL mechanisms for event registration, event publishing and event notification. The only exception from this is the registration for notifications on new documents because for this references to entities of the IHE XDS information model need to be placed into the registration message. Therefore, a mapping will be provided from the respective IHE DSUB message onto the CREDENTIAL Wallet service for subscribing to an event.

5.8.4 IHE XUA: Cross-Enterprise User Authentication

IHE XUA is a profile on the OASIS SAML standards that defines how SAML identity assertions shall be built and exchanged in a healthcare setting.

For the CREDENTIAL eHealth pilot, the normative constraints on SAML as defined by XUA will be followed. This mainly effects the encoding of identity attributes which are required for discovering the organizational affiliation and role of a user.

5.8.5 IHE APPC Advanced Patient Privacy Consent

The IHE APPC integration profile defines how XACML policies shall be defined for matching XUA user attributes with XDS resource attributes (e.g., how to express that a user of professional role “dentist” may only access documents that were assembled by other dentists or orthodontists).

While the CREDENTIAL eHealth pilot solely relies on the CREDENTIAL access control services for protecting resources, all permissions will be encoded based on attributes as defined in IHE APPC.

5.8.6 HL7 FHIR: Fast Healthcare Interoperability Resources

The HL7 FHIR standard provides modular resource definitions (e.g., “patient”, “care plan”, “medication”) that can be easily combined to implement arbitrary complex and interoperable information objects. The standard defines these resources in a platform independent way so that they can be implemented using different standards while still being semantically interoperable. Recently HL7 provides bindings to XML and JSON for all defined resources together with REST interface definitions for managing instances of these resources.

For the CREDENTIAL eHealth pilot HL7 FHIR will be the standard of choice for encoding medical data that is shared through the patient’s Personal Health Record (PHR):

- Data gathered by the patient is transformed to HL7 FHIR resources before being sent to the PHR
- Medical documents assembled by doctors are defined as sets of FHIR resources within a document wrapper
- All data retrieved from the PHR is provided through FHIR resources which allows to re-used existing data rendering libraries for displaying that data in a user-friendly manner

5.8.7 Terminologies and Value Sets

In the healthcare domain, any semantic information is usually coded through one or more concepts. For instance, if a doctor wants to express that a patient is suffering from diabetes he does not just write “diabetes” into a doctor’s letter but uses a coded value that even allows expressing details about the patient’s disease (e.g., type of diabetes). Such coded values are defined through terminologies which define lists, hierarchies or even networked ontologies of concepts. For example, for coding diseases the ICD-10 terminology or the more fine-grained Alpha-ID terminology can be used which both define concepts on several hundred diseases.

International terminologies often include thousands of concepts. For an eHealth solution one usually wants to restrict the value range of a coded value to a defined subset of the codes available in a terminology. E. g. the CREDENTIAL eHealth pilot will only consider a small range of data from personal health devices such as pulse, body weight and steps count. Messages shared among IT-Systems shall only contain such data and therefore only the respective concepts for these data shall be allowed within CREDENTIAL messages.

A common practice in healthcare IT is therefore to define dedicated value sets as sets of concepts which were taken from defined code systems. Value sets can either be defined by listing all concepts to be included with the set (extensional definition) or by defining a query that results in the concepts to be included with the set (intensional definition).

5.8.7.1 LOINC (2.16.840.1.113883.6.1)

LOINC (Logical Observation Identifiers Names and Codes) is a widely used universal terminology for tests, measurements, and observations. LOINC coded values usually not only define what was measured but even how this was done (e. g. codes for body weight allow to differentiate on whether the weight was measured or guessed, with or without clothes, measured by the patient or a doctor, etc.).

For the CREDENTIAL eHealth pilot LOINC will be used for

- semantically classifying observations and measurements, e.g., signaling that a certain value reflects the patient’s body weight,
- classifying documents in conformance to IHE XDS metadata definitions.

5.8.7.2 UCUM (2.16.840.1.113883.6.8)

UCUM (Unified Code for Units of Measure) defines a full catalogue of measurement units. By integrating the ISO 1000, ISO 2955-1983, ANSI X3.50-1986, and ENV 12435 standards, UCUM implements a unified system that resolves all conflicts among these standards. UCUM is closely related to LOINC as many LOINC codes define which UCUM measurement shall be used for expressing the result of an observation (e.g., body weight represented as kilo grams).

For the CREDENTIAL eHealth pilot UCUM will be used for encoding measurement units in conjunction with LOINC coded observations.

5.8.7.3 SNOMED CT (2.16.840.1.113883.6.96)

The most complete terminology in healthcare is SNOMED CT which includes more than 300.000 clinical concepts and about a million associations. In contrast to the terminologies sketched above, SNOMED CT’s license does not allow a free use but requires an IHTSDO membership. While several European countries are members of IHTSDO, others (e.g., Germany) are not which imposes severe problems when using SNOMED CT in cross-border healthcare exchanges.

Therefore, SNOMED CT will be used in CREDENTIAL if and only if the use of SNOMED CT is demanded by another standard which is indispensable for CREDENTIAL (e.g., HL7 FHIR or HL7 CDA). If such a standard is published by HL7 International (e.g., FHIR), SNOMED CT codes may be used in the specification but shall be mapped onto local standards in the implementation within countries which are not members of the IHTSDO.

5.8.7.4 HL7/FHIR/IHE Terminologies

In addition to the more general standard terminologies introduced above, CREDENTIAL will make use of several terminologies which were designed for coding single IHE XDS metadata elements.

- FHIR-practitioner-role (2.16.840.1.113883.4.642.1.251): The FHIR practitioner role terminology is used for classifying coarse grained roles within a care setting. In CREDENTIAL a subset of this terminology is used as a basis for the CREDENTIAL Author Roles value set.
- ihede-codesystem-8 (1.3.6.1.4.1.19376.3.276.1.5.8): In 2016 IHE Germany took responsibility for defining value sets to be used for coding XDS metadata in EHR solutions for the German healthcare system. The IHE Germany code system 8 lists document class codes which are not defined with the required coarse granularity in any existing terminology. In CREDENTIAL a subset of this terminology is used as a basis for the CREDENTIAL Document Class Codes value set.
- ihede-codesystem-6 (1.3.6.1.4.1.19376.3.276.1.5.6): The IHE Germany code system 6 lists document format codes for document schemes defined by IHE Germany and HL7 Germany. In CREDENTIAL a subset of this terminology is used as a basis for the CREDENTIAL Document Format Codes value set.
- ihede-codesystem-4 (1.3.6.1.4.1.19376.3.276.1.5.4): The IHE Germany code system 4 lists practice setting codes for regular care providers within the German healthcare system. In CREDENTIAL a subset of this terminology is used for the CREDENTIAL Practice Setting Codes value set.
- ihede-codesystem-5 (1.3.6.1.4.1.19376.3.276.1.5.5): The IHE Germany code system 5 lists practice setting codes which are not regular care providers within the German healthcare system but nevertheless may participate in health data communications. In CREDENTIAL a subset of this terminology is used for the CREDENTIAL Practice Setting Codes value set.
- ihedeHealthcare-facility-types-2 (1.3.6.1.4.1.19376.3.276.1.5.2): The IHE Germany code system “Einrichtungsarten der patientenbezogenen Gesundheitsversorgung” lists healthcare facility types for regular care providers within the German healthcare system. In CREDENTIAL a subset of this terminology is used for the CREDENTIAL Healthcare Facility Types value set.
- ihedeHealthcare-facility-types-3 (1.3.6.1.4.1.19376.3.276.1.5.3): The IHE Germany code system “Einrichtungsarten ausserhalb der patientenbezogenen Gesundheitsversorgung” lists healthcare facility types which are not regular care providers within the German healthcare system but nevertheless may participate in health data communications. In CREDENTIAL a subset of this terminology is used for the CREDENTIAL Healthcare Facility Types value set.
- HL7-v3-confidentiality (2.16.840.1.113883.5.25): The HL7 v3 Confidentiality code system provides different kinds of codes that control the confidentiality of a document, e.g., based on the kind of information or the access context. CREDENTIAL will only use a subset of these codes as the value set CREDENTIAL Document Confidentiality Codes.
- HL7-v3-null-flavor (2.16.840.1.113883.5.1008): This terminology defines null values which may be used for encoding mandatory elements in case that the required information is not available or may impose patient privacy concerns.

6. eBusiness Pilot

The eBusiness pilot consists mainly of a web application available to InfoCert users on the Internet which aim to provide access to InfoCert e-commerce application. One of the eBusiness pilot use case consists of an android app available to InfoCert Legalmail users which permits to the users themselves to

- ask for the generation of a re-encryption key to be sent to Legalmail server that will store and use it when a forward filter is activated;
- ask for the decryption of a re-encrypted forwarded message.

InfoCert and its users can have several benefits using CREDENTIAL components. In fact, CREDENTIAL provides a user-friendly and secure way to share eBusiness-related data, while the users remain in full control of their personal information.

The pilot will be set up in InfoCert environment. The evaluation will be performed by measuring technical KPIs.

This section will describe the scope and architecture of the pilot, the selected use cases, the expected results and the Key Performance Indexes chosen, the technologies involved in the pilot and a view of the pilot setup.

6.1 Scope

Today many business processes in many market sectors can be performed online. But there is always a trade-off between security and usability: in fact, often the online services that are simple for users do not guarantee the right level of security and privacy. On the other hand, to implement safer processes companies sacrifice usability for users. Based on the above, in eBusiness pilot we analyzed the most widespread use cases related to the technologies developed under the project: among these, there are the authentication and Single Sign-On (SSO), the purchase processes and online form subscription, the use of digital signature software and web communication tools with legal value. All with the aim to show how the new technologies developed and CREDENTIAL services can meet the needs of security and privacy but also guarantee a simple user experience.

For these reasons, we focused on setting and evaluating different use cases in which CREDENTIAL is used “as a service” or integrating specific components (e.g., re-encryption libraries).

The use cases analyzed are:

- Identity federation with CREDENTIAL account
- Re-encryption of messages within Legalmail to empower trust in legal communication
- Import Data from CREDENTIAL Digital Wallet

All these use cases have the purpose to achieve the dual goal of security and simplicity:

a) In the case of the Identity Federation, User who has a CREDENTIAL account can easily access to a service like Ecommerce without having to remember many, too many, credentials but with a very good level of security offered by the system.
b) In the case of proxy re-encryption, Legalmail user can forward an important mail to a trusted user, protecting the message itself.
c) In the case of subscription within the e-commerce, there are several advantages: in fact, customers can tap into the information stored in the wallet by having them all at hand and share them securely with the application.

Within this section we are going to detail how the use cases will be implemented, which are the expected results and, most importantly, how we will measure the results.

6.2 Architecture Description

Two environments are relevant for the eBusiness pilot, namely InfoCert's e-commerce platform and Legalmail registered email service. These environments are introduced in the following two sections.

6.2.1 InfoCert e-commerce

The InfoCert e-commerce is a Java Enterprise web application. The service is exposed by an Apache web server and is accessed by the users with a web browser. The web server communicates with the JBoss EAP 6 application server where the Java Enterprise web application is installed. The communication between the web server and the application server is handled by the mod_jk module. The data are stored in an Oracle Database and accessed by the applications using the Java persistence API, through Enterprise Java Beans. The software components are divided between a frontend and backend.

The components of the InfoCert e-commerce are described in Figure 19:

- Frontend: developed with Liferay framework and deployed in a Portlet Container
- Backend: divided in two modules, both developed with Java EE. The first module exposes REST services to the frontend. The second module is used to access the database with Java Enterprise API, through ejb.

The authentication and part of the registration is managed by another service, the InfoCert Identity Manager.

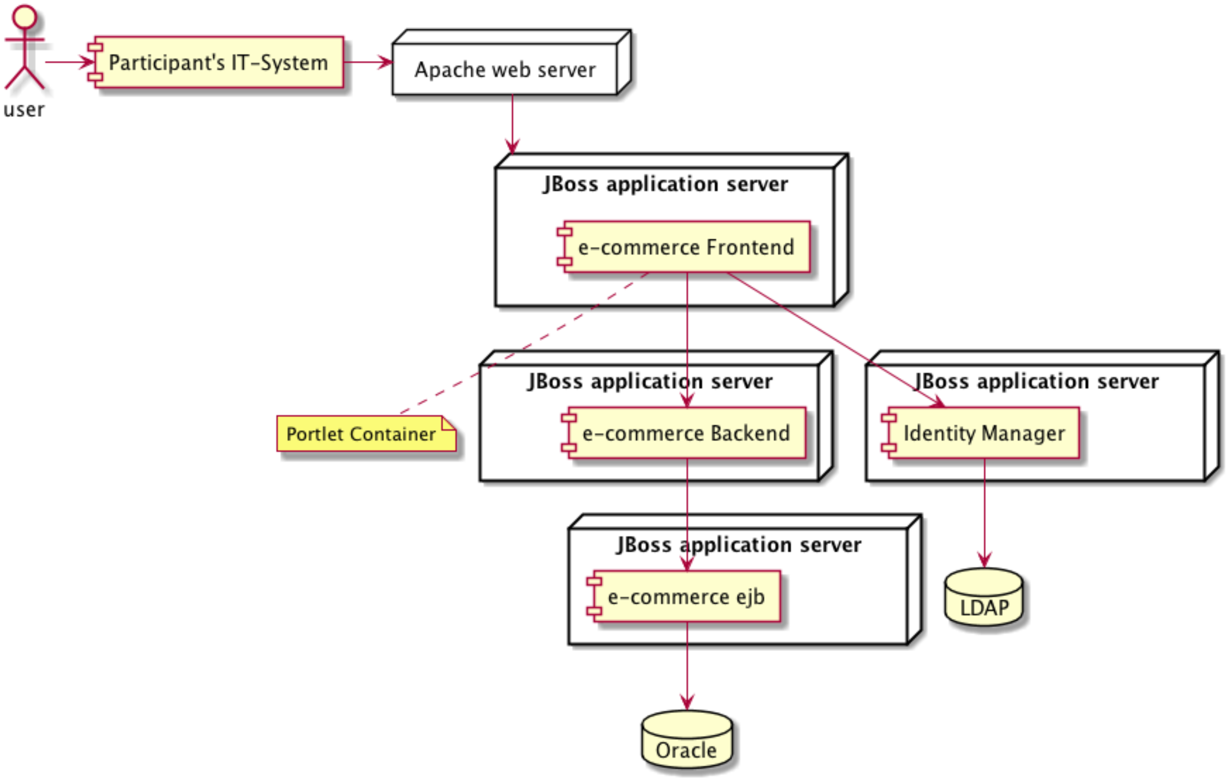


Figure 19: InfoCert e-commerce architecture

6.2.2 Legalmail registered mail service

The Legalmail mail service is implemented by components developed both in Java and C. The architecture is described in **Error! Reference source not found.**. The core of the service is the Legalmail PEC Engine, which is developed in Java. The messages handling and delivery is done by collaboration between PEC components and other SMTP servers.

Messages are stored in the File System and the other data, required for the PEC protocol compliance, are stored in an Oracle database. The messages sent through the PEC system are processed by different components depending on the two types of recipient:

- Internal to the InfoCert controlled domains: Messages are delivered to the mailbox with an SMTP backend server.
- External to the InfoCert controlled domains: Messages are sent to another PEC manager using an outgoing SMTP server.

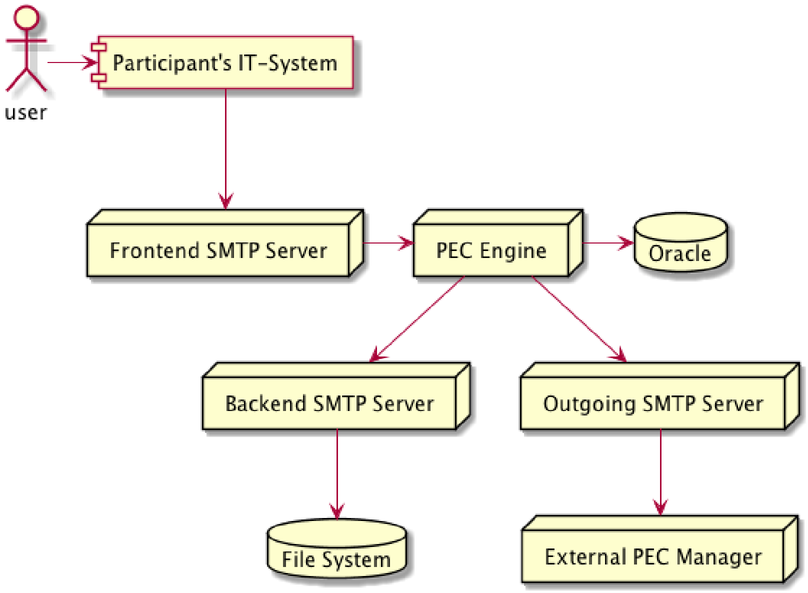


Figure 20: Legalmail registered mail service architecture

6.3 Selected Use Cases

There are three proposed eBusiness use cases. Such use cases involve some of the most important services in InfoCert business, e-commerce and Legalmail services.

1. InfoCert E-commerce: login and registration
2. InfoCert E-commerce: Legalmail contract form filling
3. Legalmail service: Legalmail encrypted message forward

6.3.1 InfoCert e-commerce: login and registration use case

In this use case the user is able to login and register to the e-commerce, creating an InfoCert account, without the need of an additional credential, being able to use the authentication and the data provided by CREDENTIAL. The use of CREDENTIAL technologies enhances the registration and login process with the added value of a strong authentication feature that nowadays is missing in the InfoCert e-commerce platform. Additionally, the creation of an InfoCert e-commerce account with the data provided by the CREDENTIAL Wallet, lets the user create an e-commerce account with a simpler process, and rely in that account also for future purchases.

6.3.2 InfoCert e-commerce: Legalmail contract form filling

The CREDENTIAL Wallet will be integrated also during the Legalmail mailbox request process. In Italy Legalmail is the registered e-mail service with legal value (PEC), and signing of a contract between the customer and the PEC manager before the mailbox creation, is a constraint defined by Italian laws. The data entered into the PEC contract shall be verified to certify that the user (natural person or legal person) is the holder of the registered email mailbox. With the integration of the CREDENTIAL the customer will be able to import his data directly from his wallet, following a process well known and giving explicit permission for the use of his data.

The integration of CREDENTIAL in these two use cases will add security and usability to InfoCert services, through strong authentication, permissions system and credentials unification.

6.3.3 Legalmail service: Legalmail encrypted message forward

The other selected use case involves directly a specific InfoCert service, Legalmail messaging. Legalmail messaging service lets the user encrypt the contents of his messages before sending them to the server for the certified delivery to the recipient, adding a valuable privacy enhancement to the certainty of the delivery (or certainty of the missed delivery) provided by the PEC protocol. However, using a standard message content encryption protocol leads to the loss of some Legalmail functionalities, one of which is the mail forward configuration. The message forward rule is still available, but the content of the encrypted message is readable only by the original recipient, which holds the secret key that can decrypt the message. Giving the secret key to another user, letting him read forwarded contents, is not an option to be considered, because would permanently compromise the original recipient’s privacy.

CREDENTIAL technologies will let us maintain the mail forward feature in the encrypted message communication without losing usability and, more importantly, without compromising the user’s secret key. With CREDENTIAL mobile application and libraries integration, Legalmail will use technologies, like proxy re-encryption, forwarding the encrypted message to a selected trusted user, which will be the only one, in addition to the original recipient, able to access readable contents. This feature is important for PEC registered e-mail users, because the Italian laws consider a delivered PEC message as accepted and seen from the moment it is saved to the user’s mailbox. If the message contains important notifications and deadlines, a user, who for any reason is not able to check his inbox for a long period, may have the need to let another trusted user see his messages, without losing confidentiality and without compromising his key.

A typical example could be a company CEO which, after a long negotiation, is waiting to receive a contract for a big deal. The CEO wants to keep confidential the messages received from the contract counterpart, but could not access very frequently his Legalmail mailbox for a certain period. The contract may require the signing before a strict deadline, so the CEO can configure a mail forward filter, towards his trusted secretary, for all the messages sent to him by the counterpart. With the CREDENTIAL feature integrated in the Legalmail system, also the secretary will receive encrypted messages and be able to read them and alert the CEO when the contract is received.

The use case will be implemented using the Android Legalmail client. Using the client the user will be able to configure mail forward filter for encrypted messages leveraging on the CREDENTIAL mobile application.

6.3.4 Selection Justification

After a deeper analysis, InfoCert decided not to implement another considered use case, the Legalmail webmail login. The decision was made after seeing strong similarity to another use case. Another strong reason for the webmail login exclusion was to put more focus on the email forwarding use case, which is more challenging and therefore needs further resources.

6.4 Details on Selected Use Cases

The use cases will be implemented with the integration of some new technologies which actually are not present in the InfoCert e-commerce and Legalmail environments.

6.4.1 InfoCert e-commerce login

The actual e-commerce authentication is made by giving proxy to the InfoCert Identity Manager, a Java Enterprise application. The user can login to the e-commerce only after a registration ending with the creation of an InfoCert account.

During the registration process the user lands with his browser in the e-commerce frontend, developed with Life ray and deployed in a Portlet Container. When the user enters its username and password in the e-commerce registration form, the credentials are passed to the Identity Manager, a Java Enterprise Application that manages InfoCert accounts. The username is stored in a LDAP, so is stored the hash of the password. After successful registration in the Identity Manager system, some additional data are stored in the e-commerce Oracle database, but no credentials are included.

The e-commerce login process is performed by the Identity Manager, after the user types in his credentials. The Identity Manager authenticates the user verifying the credentials stored in the LDAP and returns the authentication results to the e-commerce web application, which loads, if needed, some additional information from the Oracle Database.

With the CREDENTIAL integration, there will be no need for the user to follow the registration process. The first time the user will choose to login to the e-commerce using its CREDENTIAL Account, an assertion request will be composed and sent to the CREDENTIAL Wallet (e.g., SAML, OAuth 2.0) by the frontend, the request will specify the need of username and CREDENTIAL Account ID attributes. The response to the request will bring the authentication result and the information required. If the authentication is successful, the e-commerce and Identity Manager will check if an InfoCert account linked to that specific CREDENTIAL Account ID is already present. If the CREDENTIAL Account had already been stored, a session will be created and the access will be allowed to the user. It the CREDENTIAL account ID had not already been stored, an InfoCert e-commerce account will be created and saved in the Identity Manager LDAP (and in the e-commerce Oracle database) with the username and the new CREDENTIAL Account ID attributes just received. The Identity Manager and the e-commerce will be able to recognize an account created with CREDENTIAL with the link between the username and the CREDENTIAL Account ID.

In the following table, there is a description of the main steps of this use case

Steps of Ecommerce Login use case	Definition
User opens Ecommerce login page to access to the service and proceed with purchase	The ecommerce login page allows the possibility to login with other IDENTITY Systems
User chooses to use CREDENTIAL account to sign up or login	LUC: REQUEST AUTHENTICATION WITH CREDENTIAL
User logins into CREDENTIAL account system	LUC: ACCESS USING CREDENTIAL
User is redirected to purchase page logged in.	After the CREDENTIAL authentication the process continues on InfoCert ecommerce platform

Figure 21 shows the components involved in the use case and the interaction with the CREDENTIAL Wallet:

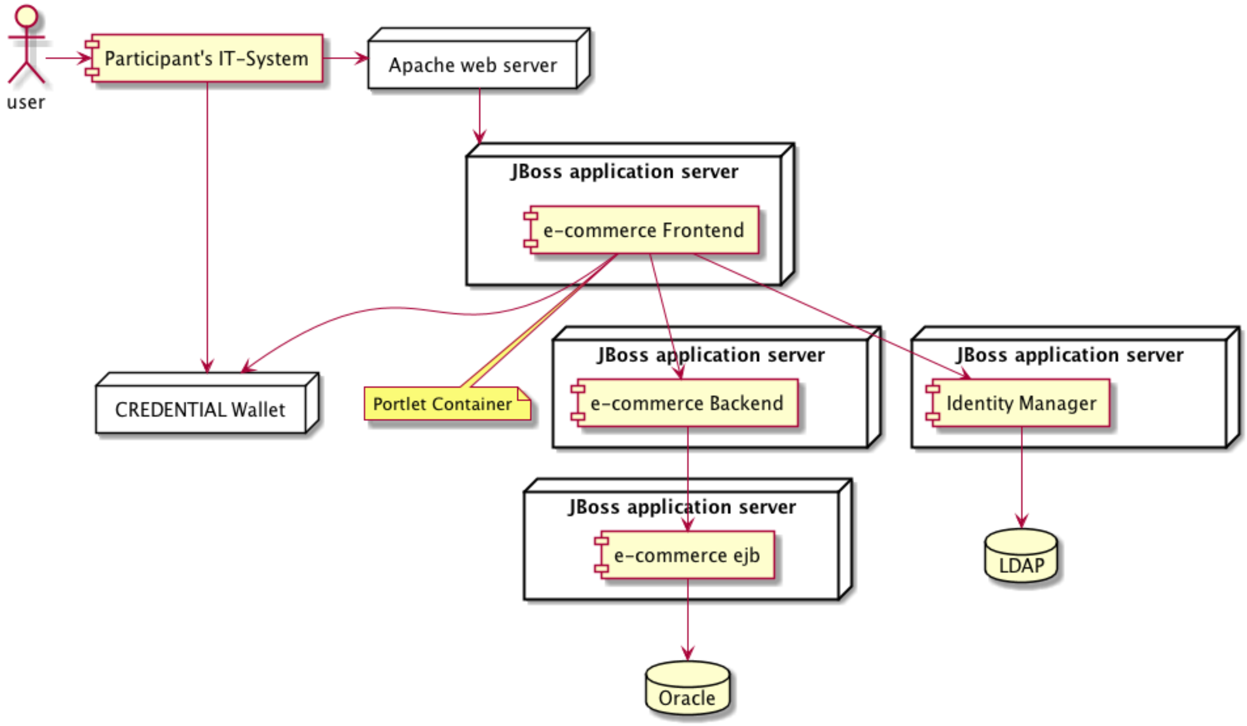


Figure 21: InfoCert e-commerce login technology description

6.4.2 InfoCert e-commerce form filling

When a user has logged in the InfoCert e-commerce web application, he can request a Legalmail mailbox. The process for a mailbox request requires a contract form to be filled. This form includes some user's personal data that can be easily imported from the CREDENTIAL Wallet. The user will navigate with his browser to the empty form, and choose to import his data from the CREDENTIAL Wallet. The e-commerce frontend will create an assertion request (e.g., SAML) requiring the attributes that will be used to fill part of the form. After receiving the attributes from the wallet, the frontend will fill the form fields that map the attributes received.

In the following table, there is a description of the main steps of this use case

Steps of Ecommerce form filling	Definition
User already authenticated with CREDENTIAL chooses to activate a Legalmail account.	The ecommerce requests to choose the name of registered email address and ask for the payment.
User needs to fulfill subscription form and selects to disclose his data from the wallet to complete registration	LUC: IMPORT DATA FOR INFOCERT SERVICES
Customer by means of a point&click procedure, provides his authorization	LUC: REQUEST DATA IMPORT FROM WALLET
User can check data inserted. Eventually can modify his information and complete the registration	LUC: FILL REGISTRATION FORM LUC: SUBMIT FILLE FORM

The following figure shows the components involved in the use case and the interaction with the CREDENTIAL Wallet:

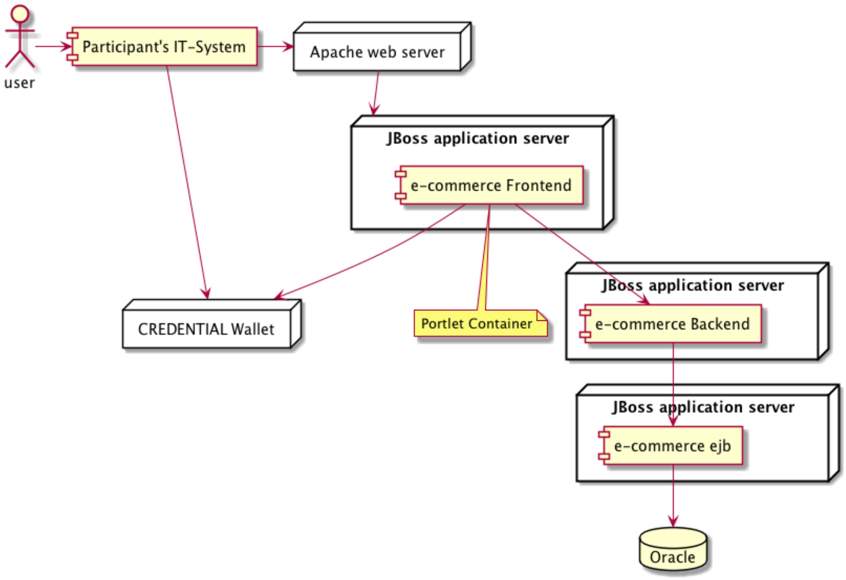


Figure 22: InfoCert e-commerce form filling technology description

6.4.3 Legalmail registered mail service

The integration of the CREDENTIAL technologies will be realized on the client side by the collaboration between the Android Legalmail Client and the Android CREDENTIAL mobile application. On the server side the Legalmail core component will use the CREDENTIAL Java libraries for the re-encryption process.

Without a mail forward rule configured, the content of the messages managed in this use case are encrypted following the S/MIME protocol. In S/MIME, the body of a MIME message is encrypted with symmetric encryption. The symmetric key is encrypted with the recipient public key and put in the enveloped data part of the MIME message. These operations are performed by the Android Legalmail client, both for encryption and decryption. The PEC Engine has no need to edit the S/MIME encrypted messages.

The CREDENTIAL feature will enhance the Legalmail encrypted messaging by letting the user configure a mail forward filter towards a trusted person which will be able to read the contents of the encrypted message.

This use case can be divided in three phases, each one using different technologies:

1. Mail forward filter configuration
2. Message re-encryption
3. Message visualization

In the mail filter configuration phase, when the user has established an encrypted content communication with another user, he uses his Android Legalmail client to configure a mail forward filter toward a third person for the messages received by that specific sender. The Android Legalmail client will send an Intent to the Android CREDENTIAL mobile application, requesting to generate the re-encryption key and sending the mail forward recipient public key. The CREDENTIAL mobile application will retrieve the requesting user private key, generate the re-encryption key and returning it to the Android Legalmail client. The Android Legalmail client will call a PEC Engine REST service to send the data containing forward rules and re-encryption key for the filter configuration. The PEC Engine will receive the filter data, save them in Sieve mail filtering language and store the re-encryption key in a key store (LDAP).

In the second phase, the PEC Engine will manage an encrypted message for the user that has configured a mail forward filter. The PEC Engine reads the PEC Envelope (enveloped following the PEC protocol) and delivers it to the original recipient. When the PEC Engine will process the Sieve mail filter, it will recognize the need of a re-encryption. The PEC Engine will extract the attachment part of the MIME PEC message, the original message. The enveloped data containing the asymmetrical encrypted symmetric key for the contents will be extracted from the MIME message, and, using the CREDENTIAL libraries, the key will be re-encrypted. The PEC Engine follows the same S/MIME and PEC protocols to put the re-encrypted symmetric key in the message structure and to create a new PEC Envelope to be sent to the forward filter recipient.

In the third phase, a forward recipient loads a forwarded re-encrypted message using his Android Legalmail client. The Android Legalmail client will load the re-encrypted message from the IMAP server and process its MIME parts. Once the original message is extracted from the PEC Envelope, the client will get the re-encrypted symmetric key from the enveloped data, following the S/MIME protocol. The re-encrypted symmetric key will be sent with an intent to the CREDENTIAL mobile application, which will decrypt and return it in plain text. The Legalmail client, having the symmetric key, will be able to decrypt the body of the MIME message and to display it for the user.

In the following table, there is a description of the main steps of this use case:

Steps of Legalmail encrypted forward	Definition
User already authenticated within Legalmail service activates message forward towards one or more trusted users sending the re-encryption keys generated by Legalmail mobile client	LUC: ACTIVATE MESSAGE FORWARD RULE
Legalmail user sends, using Legalmail mobile client, an encrypted message to another Legalmail user	LUC: COMPOSE ENCRYPTED MESSAGE LUC: SEND ENCRYPTED MESSAGE
Trusted receiver accesses and decrypts the forwarded message with the Legalmail mobile client	LUC: RECEIVE FORWARDED RE-ENCRYPTED MESSAGE

Figure 23 explains how the InfoCert and CREDENTIAL components are involved in the use case:

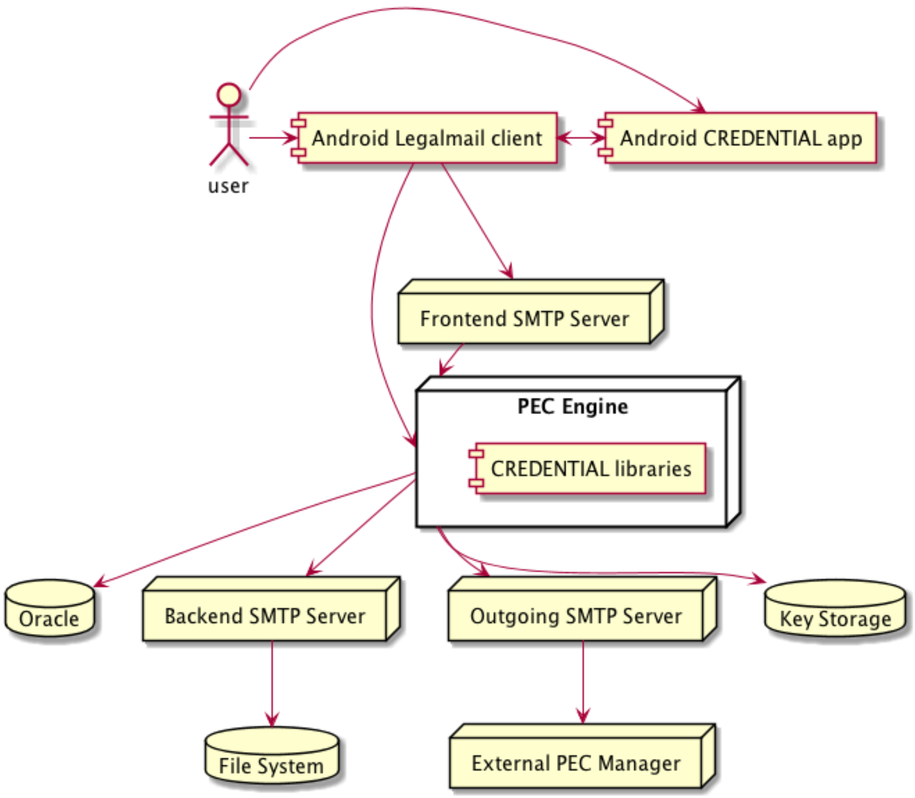


Figure 23: Legalmail registered mail services technology description

6.5 Expected Results

As described in Deliverable D2.1 “Use Cases and Scenarios”, the value proposition for the eBusiness is to demonstrate how the integration of InfoCert services with CREDENTIAL technologies can protect the users’ sensitive information and simplify certain actions during the users experience with InfoCert’s web applications. Specifically, using the Wallet components we would like to demonstrate how useful is this tool to store and share privacy sensitive information. Moreover, experimenting CREDENTIAL re-encryption feature we want to add value to InfoCert solution, enhancing the security and privacy in sharing messages with a certified email system.

For these reasons the pilot will be conducted with the aim to demonstrate these added values for the users.

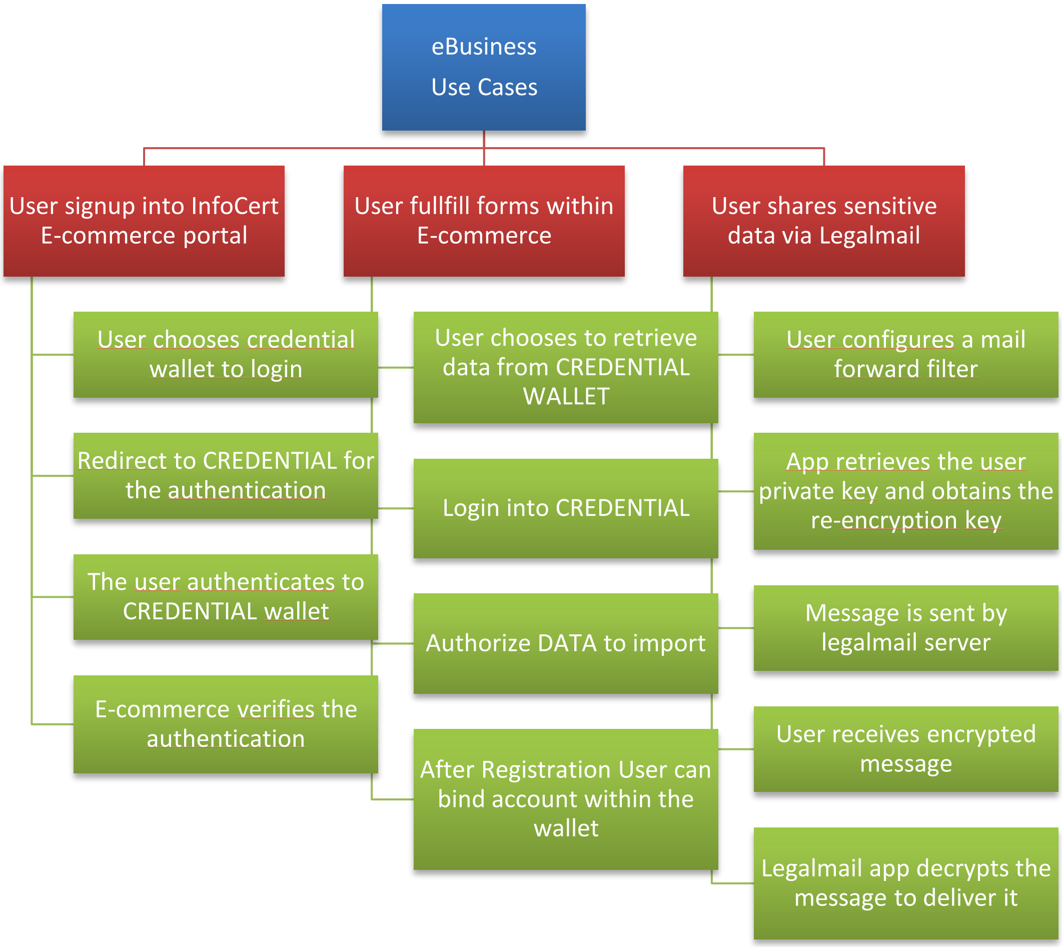

eBusiness Use Cases
User signup into InfoCert E-commerce portal
User fullfill forms within E-commerce
User shares sensitive data via Legalmail
User chooses credential wallet to login
Redirect to CREDENTIAL for the authentication
The user authenticates to CREDENTIAL wallet
E-commerce verifies the authentication
User chooses to retrieve data from CREDENTIAL WALLET
Login into CREDENTIAL
Authorize DATA to import
After Registration User can bind account within the wallet
User configures a mail forward filter
App retrieves the user private key and obtains the re-encryption key
Message is sent by legalmail server
User receives encrypted message
Legalmail app decrypts the message to deliver it

Figure 24: Selected Use Cases of the eBusiness Pilot

6.5.1 User Signup with CREDENTIAL account within InfoCert e-commerce

In this scenario, we expect to create the conditions under which the users can avoid a long registration process to subscribe trust services via web. Leveraging on CREDENTIAL Wallet, the user can reduce the time needed for registration, enforce the privacy of his data, avoid remembering the umpteenth password. Moreover, at the moment the e-commerce requires just the username and password to login into the system. After the integration, the user will be able to login to the e-commerce using the strong authentication features provided by the CREDENTIAL system.

6.5.2 User fulfill forms within e-commerce retrieving data from wallet

To complete subscription on a trust service (like Legalmail PEC), the user needs to register his data in a request form that should be signed and shared with InfoCert to complete the purchase and proceed with the service activation.

Retrieving data from wallet we expect to reduce the bounce rate in this step of process, to lower the risk perception in sharing information and the rate of errors in fulfilling forms. After this demonstration, a possible future evolution could be to include even more sensitive data during the InfoCert services registration process, like the credit card number during the payment step.

6.5.3 User forwards important messages with Legalmail using re-encryption feature.

In the Legalmail encrypted message case the user will be able to maintain encrypted his message’s sensitive contents and let a trusted person check important communications. This is an important feature because of the legal value of the Italian Registered email system. In some cases, a user is required to check its certified mailbox and respect deadlines for which the starting date is the day of the message delivery. Giving to a trusted person the control over some messages let the original recipient fulfill the legal constraints imposed by the Italian Registered email system. With the features provided by the CREDENTIAL mobile application and libraries, InfoCert PEC environment will let the user designate a trusted person to read specific encrypted messages, through a simple mail forward filter configuration. These operations are done without the need to disclose the original recipient’s private key.

6.5.4 Summary of Expected Results

Expected Result	Acceptance Criteria
User Signup with CREDENTIAL account within InfoCert e-commerce	- Customer must not be register to the ecommerce to complete a purchase - User signup to the e-commerce with CREDENTIAL account
User fulfill forms within e-commerce retrieving data from wallet	- User shall not type all information in the various labels - User can avoid to store information within the browser
User login into Legalmail account	- User can login into Legalmail without typing is Legalmail password - User can login with CREDENTIAL strong authentication
User forwards important message with Legalmail using re-encryption feature	- User can forward encrypted messages with Legalmail mobile app to another customer without sharing his Private Key - Recipient can open the mail forwarded containing the decrypted message

6.6 Key Performance Indicators Used to Evaluate the Pilot

To better analyze the effects on customers behavior in using InfoCert online services introducing CREDENTIAL technologies, we decided to measure both quantitative and qualitative Key performance indicators: thanks to quantitative indicators we are going to understand if CREDENTIAL enhances the application performances in terms of improved security and speed of process; on the other hand, we want evaluate customers satisfaction about these new features and better understand the perceived value. To do so we will set a series of qualitative analysis. At the end of the analysis we would like to demonstrate:

- The improvement of security;
- The speed of the process;
- The easiness of the process;
- The perceived value;

- The customer satisfaction.

6.6.1 Quantitative KPI

For each use case, we will try to measure some quantitative indicator that could show the improvements in terms of user experience.

6.6.1.1 InfoCert E-commerce: login and registration

Description	We need to keep an acceptable waiting time for the communication between the e-commerce and the wallet when a “Login with CREDENTIAL is required”
What will be measured	The time from when the user click on the link “Login with CREDENTIAL” and the landing on the wallet login page
How it will be measured	The timing will be measured in milliseconds
Reasoning	The communication between the e-commerce and the wallet will be implemented with a protocol which will affect the user’s navigation. The user will have to perform some actions using the browser (e.g. a window opens and asks for user and password), but behind the scenes, there is a communication and validation done by the e-commerce and the wallet, this process must be efficient and as fast as possible.
Success Scenario	The waiting time between the user clicking on “login with CREDENTIAL” link in the e-commerce and the wallet’s login window less than 2000 ms.

Description	We need to keep an acceptable waiting time for the communication between the wallet and the e-commerce when the user has given permission to export from the wallet his e-mail and CREDENTIAL ID to the e-commerce
What will be measured	The time from when the user gives permission to export required data and the landing on the e-commerce page.
How it will be measured	The timing will be measured in milliseconds
Reasoning	The communication between the wallet and the e-commerce will be implemented with a protocol which will affect the user’s navigation. The data export in this process must be efficient and as fast as possible.

Success Scenario	The waiting time between the user giving permission to export the required data and the landing on the e-commerce first page is less than 2000 ms.

Description	We need to keep an acceptable waiting time for the communication between the e-commerce and the wallet when a “Import data from CREDENTIAL Wallet is required”
What will be measured	The time from when the user clicks on the link “Import data from CREDENTIAL Wallet” and the landing on the Wallet page
How it will be measured	The timing will be measured in milliseconds
Reasoning	The communication between the e-commerce and the wallet will be implemented with a protocol which will affect the user’s navigation.
Success Scenario	The waiting time between the user clicking on “Import data from CREDENTIAL Wallet” link in the e-commerce Legalmail form section and the wallet page is less than 2000 ms.

Description	We need to keep an acceptable waiting time for the communication between the wallet and the e-commerce when the user has given permission to export from the wallet his e-mail and CREDENTIAL ID to the e-commerce service
What will be measured	The time from when the user give permission to export required data and the landing on the e-commerce form page, with the form partially filled.
How it will be measured	The timing will be measured in milliseconds
Reasoning	The communication between the wallet and the e-commerce will be implemented with a protocol which will affect the user’s navigation. The data export in this process must be efficient and as fast as possible.
Success Scenario	The waiting time between the user giving permission to export the required data and the landing on the e-commerce form page with the form partially filled is less than 2000 ms.

Description	We will verify how many data the user can import from its wallet in terms of number of form fields.
What will be measured	The number of form’s fields filled with data

	imported from the wallet
How it will be measured	Counting the form's fields filled with data imported from the wallet
Reasoning	A user must have a benefit from choosing to follow the process of data import. If the fields of imported data are few, the user could prefer filling them by hand.
Success Scenario	At least five form's fields are filled with data imported from the wallet.

6.6.1.2 Legalmail service: Legalmail encrypted message forward

Description	The re-encryption key generation will be done during the mail forward filter configuration process. The key generation is performed by the CREDENTIAL mobile app and exchanged with the Legalmail client with inter App communication and the filter configuration is sent to the server with the attached re-encryption key. We will measure the timing needed for key generation and filter storage in the server
What will be measured	Waiting time between the user saving the filter configuration and the server giving a success result for the filter storage.
How it will be measured	The waiting time will be measured in milliseconds
Reasoning	A mail forward filter shall be saved with acceptable waiting time for the user. The mail filter configuration should not be a frequent operation, but should not compromise usability.
Success Scenario	The user clicks on the save mail forward filter and waits less than 5000 ms before obtaining a successful response from the server

Description	The server will use the CREDENTIAL libraries to re-encrypt the message to be forwarded after retrieving the re-encryption key from the key storage system.
What will be measured	The timing for the re-encryption process performed by the server
How it will be measured	The timing will be measured in milliseconds
Reasoning	The time to perform the re-encryption shall be measured and shall not compromise the PEC management system performances.
Success Scenario	The server is able to re-encrypt the session key of the encrypted message in less than 30000 ms

Description	The re-encryption process performed in the server will need some JVM memory to be performed. We must put some limit on the memory needed for the re-encryption process
What will be measured	The quantity of JVM memory needed during the re-encryption process
How it will be measured	The quantity of memory will be measured in MB
Reasoning	The PEC management system shall perform some critical operation for all the PEC messages. The re-encryption requested for a forward rule shall not compromise other PEC messages management
Success Scenario	The re-encryption process needs less than 20 MB of JVM memory usage.

Description	The Legalmail mobile app communicates with the CREDENTIAL mobile app to be able to obtain the key to decrypt the contents of the message.
What will be measured	The waiting time from the "Open encrypted message" click and the plain text visualization
How it will be measured	The waiting time will be measured in ms
Reasoning	A user that wants to read an encrypted message expects the operation to have a timing similar to the reading of a plain text message
Success Scenario	The user chooses to open an encrypted message and the waiting time for the visualization of the message on the display is less than 3000 ms

6.6.2 Qualitative KPI

The easiness of the processes, the perceived value of the introduced features and the customers satisfaction are of course qualitative *KPIs*. With the aim of assessing an objective evaluation about these metrics, we are going to leverage on several usability tests that will involve different kind of users.

All the tests will be supported by two different tools to measure the above KPI*s*:

- **Survey**: The users will evaluate their experience through some surveys specific for evaluating the usability of a web application, in terms of easiness, clarity, speed, added value, perception of time spent and overall satisfaction. Specifically, each question within the survey will have a rating from 1 (negative feedback) to 5 (very positive feedback), plus a blank space to give tips and describe the own perception.
- **Interview:** After performance tests, UX experts will interview each participant in relation with their own performance observed. From the post-interview analysis, we will try to create different clusters based on all the answers and perceptions:

- **Satisfaction area:** all the elements that have been perceived as positive during tasks
- **Improvement area**: all the steps and action that could be simplified or clarified.
- **Negative area:** include all the obstacles observed by the users, considered useless for the action
- **Confusion area:** all the elements that are not clear to the users, and create confusion about the purpose and actions to be performed.

The outputs will be a detailed report in which the UX experts will describe:

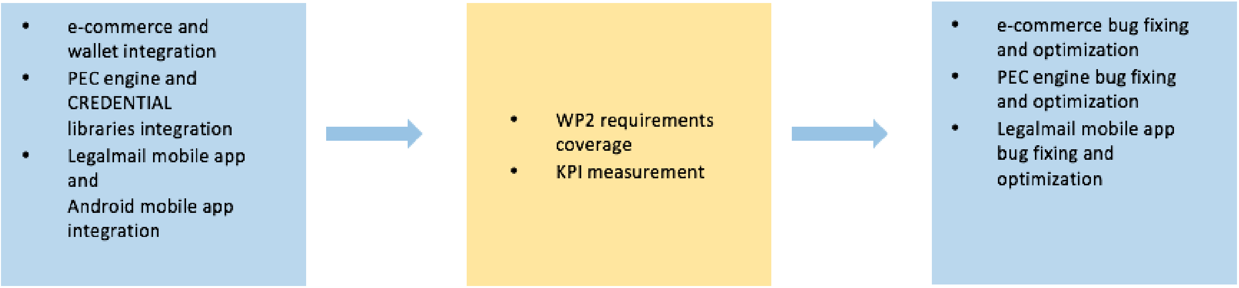
- The elements that have determined the success of the tasks
- The elements that have determined the failure of the tasks
- A list of specific errors occurred during performances
- Key outs to correct errors and improve usability

6.7 Pilot Execution

The pilot will be executed within a test environment exposed on Internet that will be hosted by InfoCert. This test environment will be a replication of the latest InfoCert's production environment. For what concerns the encrypted message forward there will be, in addition to the server side application, a client side mobile application. The Android Legalmail application used for the pilot will be a CREDENTIAL specific prototype, and will communicate with the CREDENTIAL generic mobile application and the test PEC server.

For both the server side application and the mobile application, the implementations required for the pilot will be developed within a branch started from the latest version of the production's software.

Before involving the test users, there will be a two-phases internal validation, that will let InfoCert reach a mature implementation on which the usability tests will be performed. The first phase of the execution will be focused on verifying the coverage of the requirements defined during WP2. During the first phase, there will be also the measurement of the KPIs defined in Section 6.6. From the reporting made during the first execution, we will have an input to plan optimization and we expect the need of some bug fixing activity. At the end of the second phase, there will be another test session to validate the coverage of the requirements and to measure the KPIs.



- e-commerce and wallet integration
- PEC engine and CREDENTIAL libraries integration
- Legalmail mobile app and Android mobile app integration

- WP2 requirements coverage
- KPI measurement

- e-commerce bug fixing and optimization
- PEC engine bug fixing and optimization
- Legalmail mobile app bug fixing and optimization

Figure 25 Pilot Execution steps

The pilot will be conducted through different group sessions during which the users will be asked to do some tasks and conclude certain actions.

We will select two different panel of users:

- **The first** will be represented by millennial target: they will be around twenty 18-25 years old students from university, male and female. These students will be familiar with technology.
- **The second** group will be more inhomogeneous: it will include adult and old-age people, both skilled and non-familiar with computers and internet, male and female.

Moreover, we will also involve designers and usability experts to conduct one of the usability tests scheduled.

About the analysis, the qualitative analysis will be conducted leveraging on usability test methodologies. The usability evaluation techniques are borrowed by psychological theories, ethnographical methodologies and social studies.

To evaluate the usability, we will use different techniques like:

- **Pluralistic Walkthrough**: This method includes three types of evaluators: designers, usability experts and end-users. Everyone should accomplish a list of tasks assuming the end-user's role. At the end, it is required a detailed rating for every task.
- **Usability Tests (Simple and ecological observation, thinking aloud):** Usability tests could help evaluating the deviation from the designer's logic to users logic. It's possible to measure the performance (timing and counting errors) and highlight the bottlenecks. It is possible to ask the subject to accomplish a given task (it could be a single or a list of) or simple exploring the UI under the look of the examiner, who takes notes, trying to be marginal as possible. The test could lead in a special scenario (simple observation) or in the subject's natural environment (ecological observation), to better understand the real use of the end-user. It could be useful a deeper analysis with the thinking aloud method, in which the subject is asked to speak about their thoughts on the UI. In this way, it is not possible to take the time (because the test is slower for its nature).

At the end of the test we will collect key data of the participants, their skills and habits and a personal interview and/or a survey on their single test (eg: to clarify their behavior).

The quantitative analysis will be conducted during task performance measuring all the KPIs described before and using Web Analytic Tool (e.g., Google Analytics) to measure results.

In Figure 26, the timeline regarding how the pilot will be conducted for each target is explained.

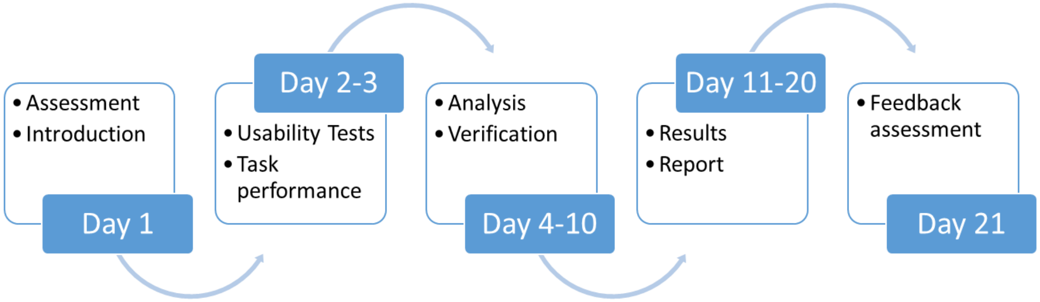


Figure 26: eBusiness evaluation time plan

7. Conclusion

The document described the technical integration of each pilot with the CREDENTIAL architecture and their components and thus gives a detailed specification for each pilot. The technical integration is given by UML sequence diagrams and a detailed explanation.

A framework for the pilot execution in form of a two-phase approach was presented. Each pilot described how they will follow this framework together with slight modification in order to take into account pilot specific requirements for the execution as well for the evaluation. Each pilot presented a list of expected results which will be evaluated at the end of each pilot phase. In order to have quantifiable and quantitative measurements, each pilot defined a set of key performance indicators which will be monitored during the pilot phase. Every pilot presented a time plan for the execution and give reasoning how and to what extent user participation is performed.

The work in this document is the starting point for the development and implementation of the generic parts of the CREDENTIAL Wallet as well as the UI interfaces for the mobile app. By knowing which components need to be integrated in each pilot, the priorities in the implementation tasks of the project can be set and technologies as well as standards can be selected. Furthermore, by knowing what types of data and to what extent it needs to be exchanged, the scientific work for choosing appropriate cryptographic algorithms can be continued.

Bibliography

[1] G. J. Myers, C. Sandler e T. Badgett, The Art of Software Testing, 3rd Edition, Wiley, 2011.

[2] Lewenson, M. Truglio-Londrigan e B. Sandra, Public health nursing: practicing population-based care, vol. 2nd ed., Burlington, Mass.: Jones & Bartlett Learning, 2013, p. p. 317.

[3] S. Smyth e A. Heron, Diabetes and obesity: the twin epidemics, Nature Medicine, 2006, p. 75–80.

[4] Diseases, National Institute of Diabetes and Digestive and Kidney, Causes of Diabetes, 2014.

[5] H. Tfayli e Arslanian, Pathophysiology of type 2 diabetes mellitus in youth: the evolving chameleon, 2009.

[6] Williams, Textbook of endocrinology, vol. 12th ed., Elsevier/Saunders, p. 1371–1435.

[7] Garg, Kelly, Voelmle, Ritchie, Gottlieb, McFann e Ellis, Improved glycemic control with real-life use of continuous glucose sensors in adult subjects with type 1 diabetes, Diabetes Care, 2007, p. 3023–3025.

[8] A. Hossaina, F. Yamaguchia, T. Matsuob, I. Tsukamotoc, Y. Toyodad, M. Ogawae, Y. Nagataf e M. Tokuda, «Rare sugar d-allulose: Potential role and therapeutic monitoring in maintaining obesity and type 2 diabetes mellitus,» *Pharmacology & Therapeutics,* vol. 155, p. 49–59, November 2015.

[9] elastic, «https://www.elastic.co/products/kibana,» [Online]. Available: https://www.elastic.co/products/kibana.

[10] IHE, «http://ihe.net/Technical_Frameworks/#IT,» [Online]. Available: http://ihe.net/Technical_Frameworks/#IT.

[11] J. Nielsen, «Why You Only Need to Test with 5 Users,» [Online]. Available: https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/.

[12] A. Crispin e L. Gregory, Agile Testing: A Practical Guide for Testers and Agile Teams, Addison-Wesley Professional, 2009.

[13] Laboratories, RSA, «PKCS 11 Base Functionality v2.30: Cryptoki - Draft 4».

[14] R. Maxim e B. Pressman, Software Engineering: A Practitioner’s Approach, vol. 8th Edition, McGraw-Hill Education, 2014.

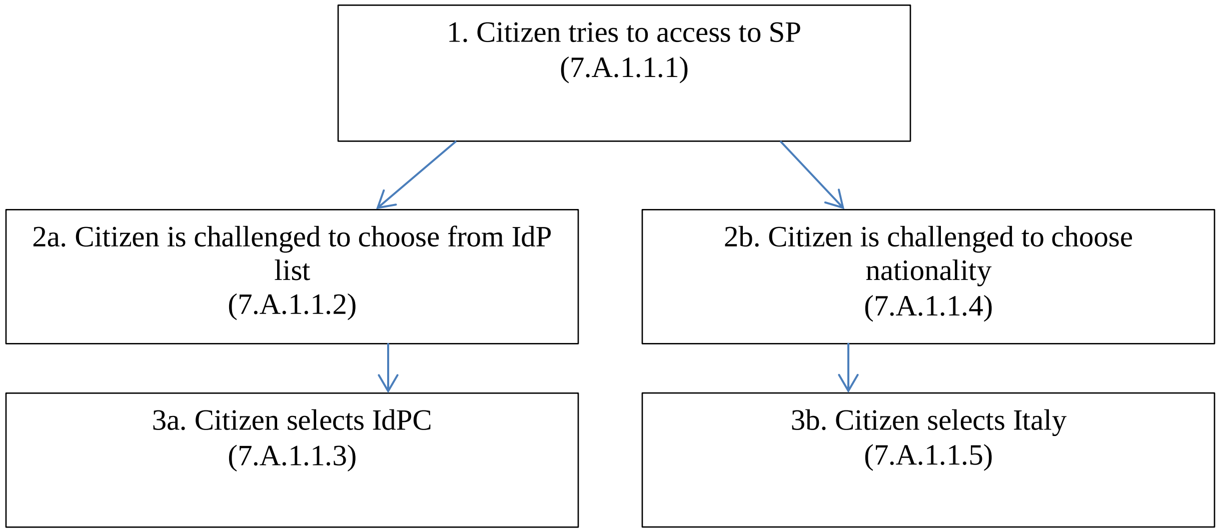
A eGovernment Use Cases

The following paragraph explains the logical and technical use cases for the eGovernment pilot.

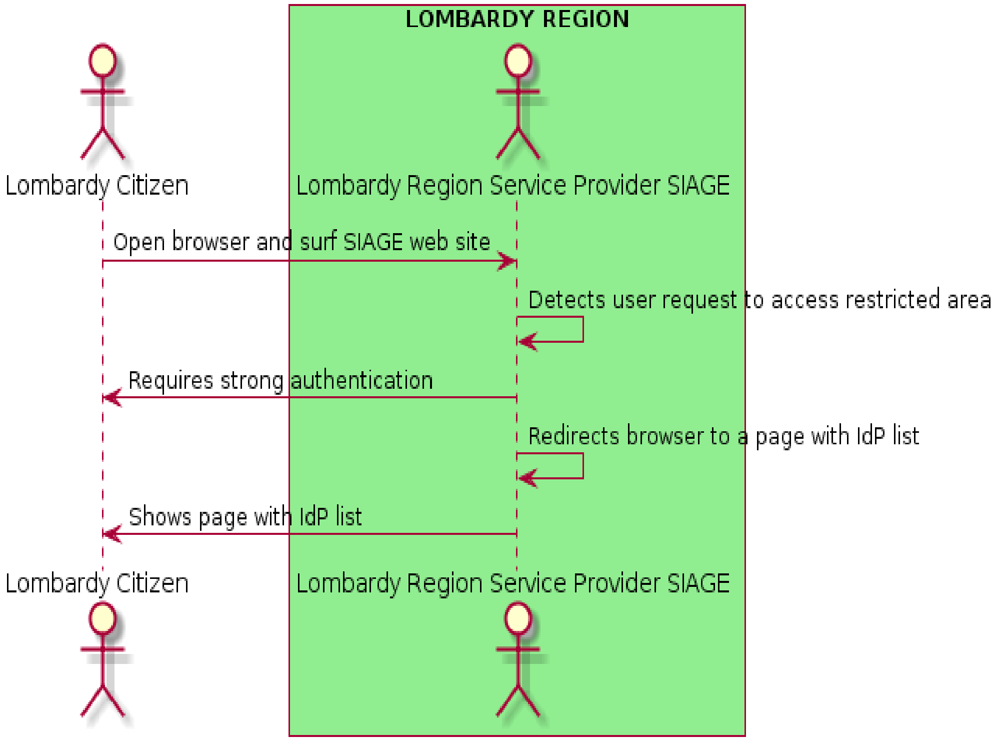
A.1 Logical Use Cases

A.1.1 BUC Citizen asks for a contribution from Lombardy Region

The flow of this BUC is “dual” - the scenario could be one of the following

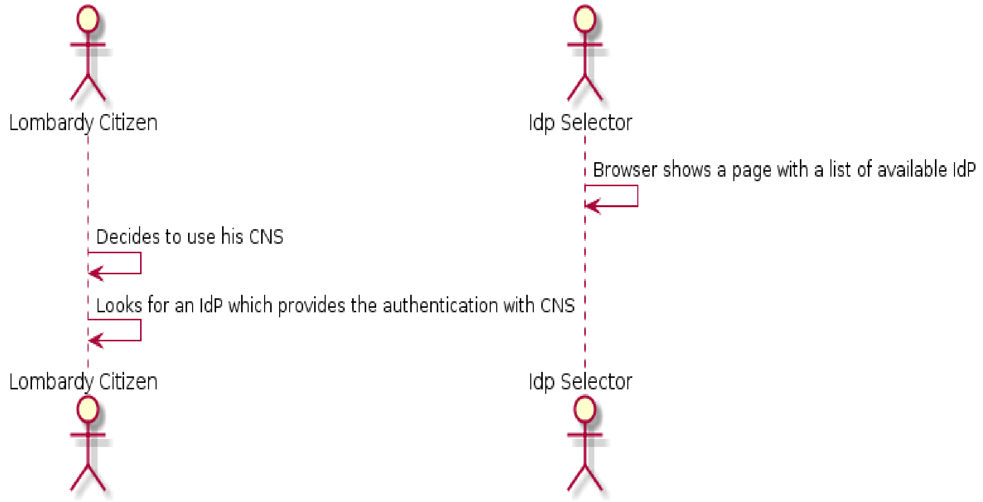

1. Citizen tries to access to SP
(7.A.1.1.1)
2a. Citizen is challenged to choose from IdP list
(7.A.1.1.2)
2b. Citizen is challenged to choose nationality
(7.A.1.1.4)
3a. Citizen selects IdPC
(7.A.1.1.3)
3b. Citizen selects Italy
(7.A.1.1.5)

A.1.1.1 1. Citizen tries to access to SP

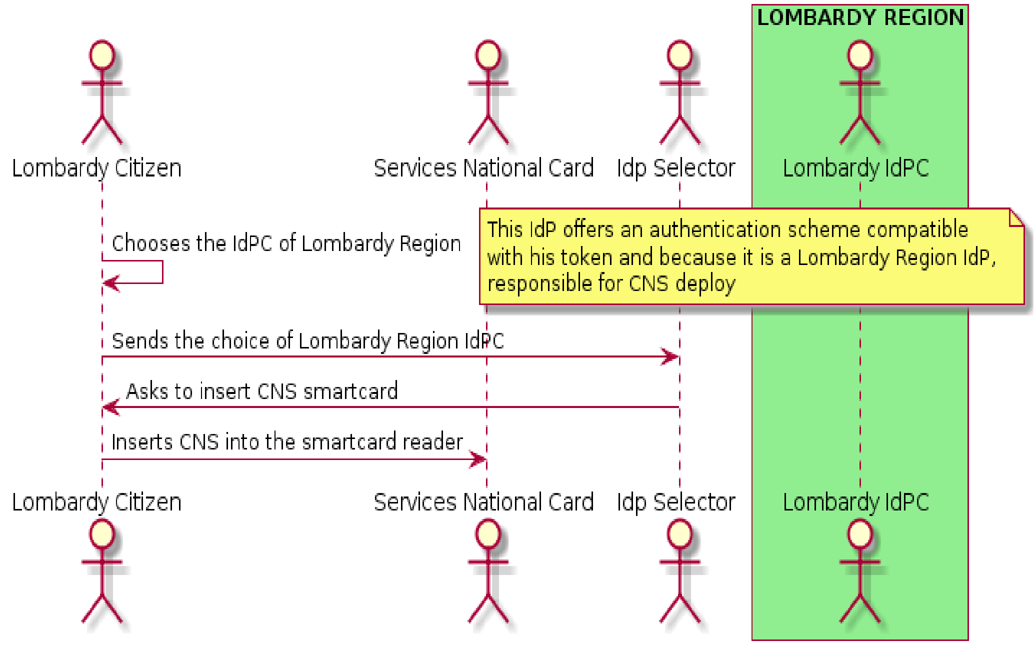
Use Case Name	**1. Citizen tries to access to SP**
ID	E.BUS-LUC-256
Main Actor	- Citizen
Secondary Actors	- Lombardy Region Service Provider -
Pre-conditions	- User needs some service offered by SIAGE web site, and has his CNS smartcard and PIN
Post-conditions	- SIAGE web site challenges Antonio to perform a strong authentication with smartcard and PIN
Description	- User tries to access to SP (SIAGE), the web site challenges user to perform an appropriate strong authentication.
Image	

LOMBARDY REGION
Lombardy Citizen
Lombardy Region Service Provider SIAGE
Open browser and surf SIAGE web site
Detects user request to access restricted area
Requires strong authentication
Redirects browser to a page with IdP list
Shows page with IdP list
Lombardy Citizen
Lombardy Region Service Provider SIAGE

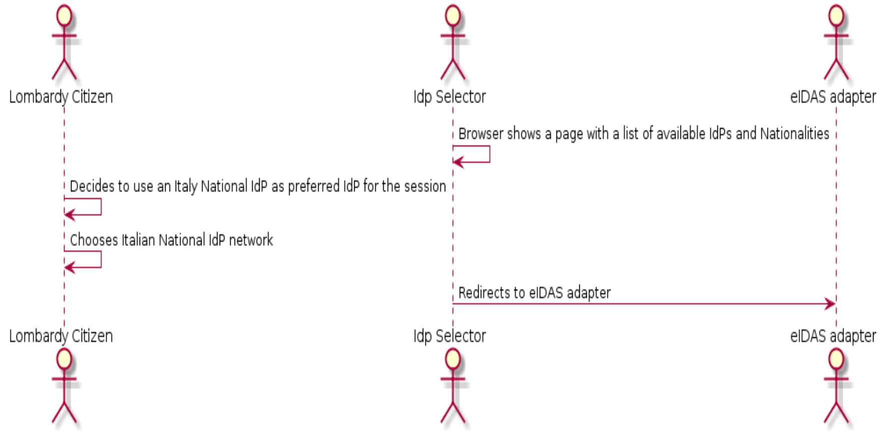
A.1.1.2 2a. Citizen is challenged to choose from IdP list

Use Case Name	**2a. Citizen is challenged to choose from IdP list**
ID	E.BUS-LUC-257
Main Actor	- Citizen
Secondary Actors	- IdP selector/adapter
Pre-conditions	- SIAGE redirects user browser to an IdP list
Post-conditions	- Antonio thinks about IdPC as preferred IdP for the session
Description	- User can choose from several IdPs available in the CREDENTIAL network. One of these is IdPC. Note that the "trigger" that forces citizen to start this behaviour is the user browser redirection to IdP selector, made by Shibboleth SP.
Image	-  Lombardy Citizen Idp Selector Browser shows a page with a list of available IdP Decides to use his CNS Looks for an IdP which provides the authentication with CNS Lombardy Citizen Idp Selector

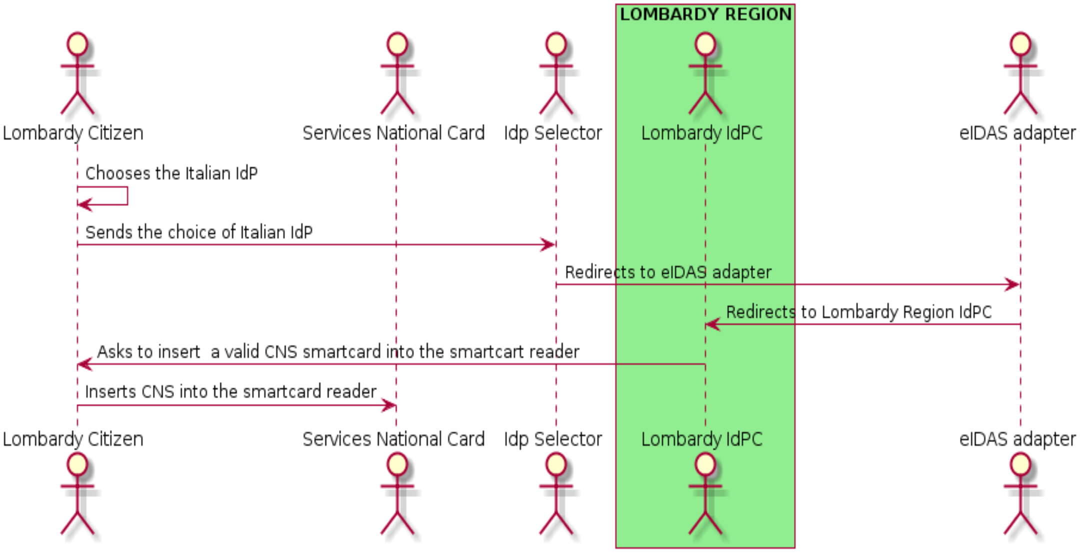
A.1.1.3 3a. Citizen selects IdPC

Use Case Name	**3a. Citizen selects IdPC**
ID	E.BUS-LUC-260
Main Actor	- Citizen
Secondary Actors	- Lombardy Region IdP (IdPC) - IdP selector/adapter - CNS
Pre-conditions	- User has an authentication token released by Lombardy Region
Post-conditions	- Antonio selects IdPC from IdP list
Description	- Citizen selects IdPC because he wants to perform a "chip and PIN" strong authentication, using his CNS.
Image	 LOMBARDY REGION Lombardy Citizen Services National Card Idp Selector Lombardy IdPC Chooses the IdPC of Lombardy Region This IdP offers an authentication scheme compatible with his token and because it is a Lombardy Region IdP, responsible for CNS deploy Sends the choice of Lombardy Region IdPC Asks to insert CNS smartcard Inserts CNS into the smartcard reader Lombardy Citizen Services National Card Idp Selector Lombardy IdPC

A.1.1.4 2b. Citizen is challenged to choose nationality

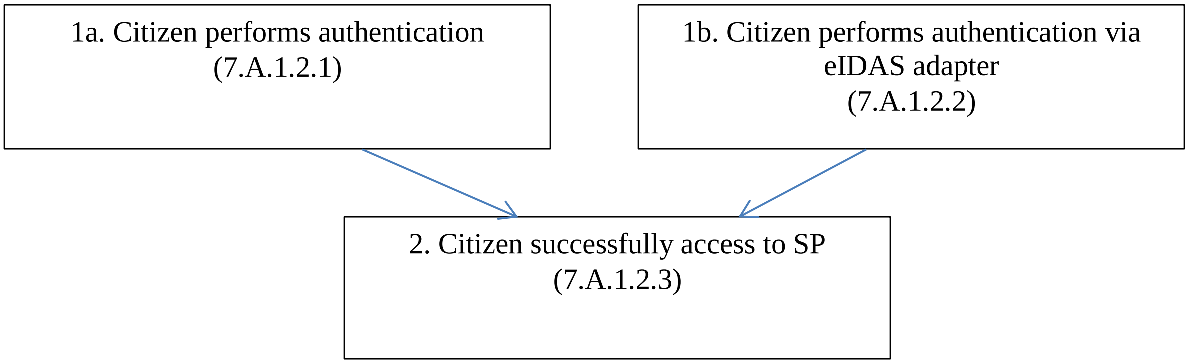
Use Case Name	2b. Citizen is challenged to choose nationality
ID	E.BUS-LUC-1544
Main Actor	- Citizen
Secondary Actors	- eIDAS adapter - IdP selector/adapter
Pre-conditions	- SIAGE redirects user browser to an IdP list
Post-conditions	- User thinks about an Italy National IdP as preferred IdP for the session
Description	- User can choose from several IdPs available in the CREDENTIAL network. He can also choose from a list of National IdP. In this use case, user chooses "Italy". Note that the "trigger" that forces citizen to start this behaviour is the user browser redirection to IdP selector, made by Shibboleth SP.
Image	 Lombardy Citizen Idp Selector eIDAS adapter Browser shows a page with a list of available IdPs and Nationalities Decides to use an Italy National IdP as preferred IdP for the session Chooses Italian National IdP network Redirects to eIDAS adapter Lombardy Citizen Idp Selector eIDAS adapter -

A.1.1.5 3b. Citizen selects Italy

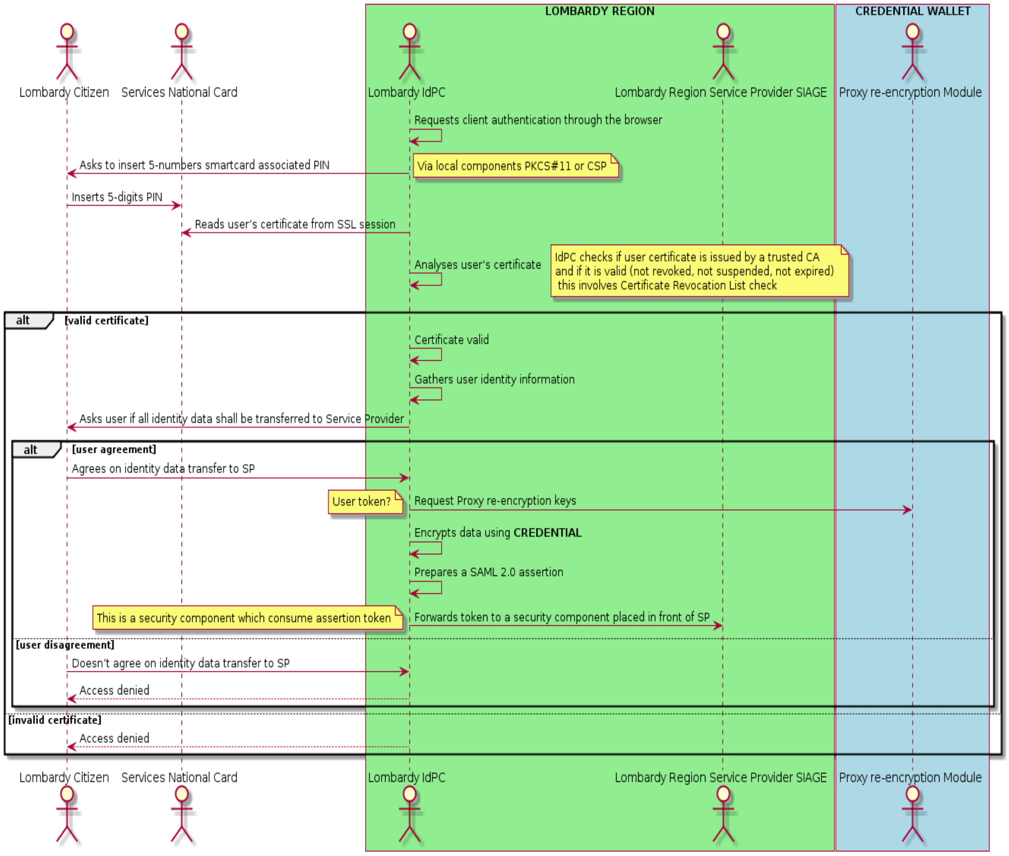
Use Case Name	3b. Citizen selects Italy
ID	E.BUS-LUC-1545
Main Actor	- Citizen
Secondary Actors	- eIDAS adapter - IdP selector/adapter - CNS
Pre-conditions	- User has an authentication token released by Lombardy Region
Post-conditions	- User selects Italy from the chooser of international Identity Provider
Description	- User chooses Italian IdP, selecting the Italian flag on IdP selector page. This redirects the user browser to IdPC. User is then challenged to perform a “chip and PIN” strong authentication, using his CNS.
Image	 LOMBARDY REGION Lombardy Citizen Services National Card Idp Selector Lombardy IdPC eIDAS adapter Chooses the Italian IdP Sends the choice of Italian IdP Redirects to eIDAS adapter Redirects to Lombardy Region IdPC Asks to insert a valid CNS smartcard into the smartcart reader Inserts CNS into the smartcard reader Lombardy Citizen Services National Card Idp Selector Lombardy IdPC eIDAS adapter

A.1.2 BUC Citizen authenticates and access to SP

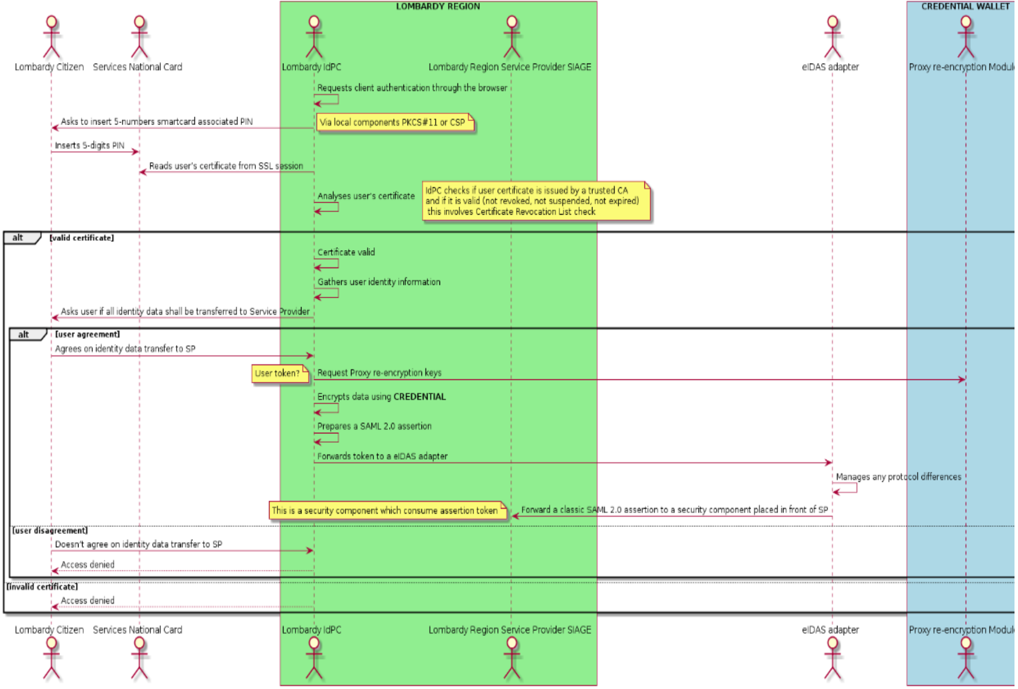
The flow of this use case is “dual” and could be either of the following two:


1a. Citizen performs authentication
(7.A.1.2.1)
1b. Citizen performs authentication via eIDAS adapter
(7.A.1.2.2)
2. Citizen successfully access to SP
(7.A.1.2.3)

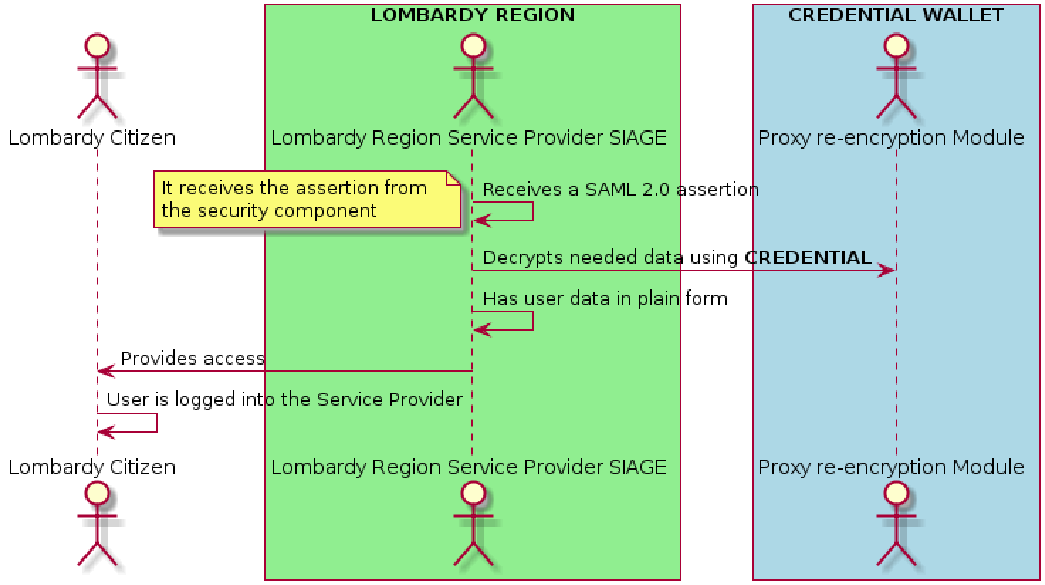
A.1.2.1 1a. Citizen performs authentication

Use Case Name	1a. Citizen performs authentication
ID	E.BUS-LUC-258
Main Actor	- Citizen
Secondary Actors	- Lombardy Region IdP (IdPC) - CNS - CREDENTIAL proxy re-encryption Module - Lombardy region Service Provider
Pre-conditions	- User has previously inserted his CNS in smartcard reader
Post-conditions	- User is successfully authenticated into IdPC
Description	- User has inserted CNS into the smartcard reader and digits the 5-numbers associated PIN. The user certificate is examined by IdPC in order to determine if it is not revoked and not suspended. Proxy re-encryption is made.
Image	 LOMBARDY REGION CREDENTIAL WALLET Lombardy Citizen Services National Card Lombardy IdPC Lombardy Region Service Provider SIAGE Proxy re-encryption Module Requests client authentication through the browser Asks to insert 5-numbers smartcard associated PIN Via local components PKCS#11 or CSP Inserts 5-digits PIN Reads user's certificate from SSL session Analyses user's certificate IdPC checks if user certificate is issued by a trusted CA and if it is valid (not revoked, not suspended, not expired) this involves Certificate Revocation List check alt [valid certificate] Certificate valid Gathers user identity information Asks user if all identity data shall be transferred to Service Provider alt [user agreement] Agrees on identity data transfer to SP User token? Request Proxy re-encryption keys Encrypts data using CREDENTIAL Prepares a SAML 2.0 assertion This is a security component which consume assertion token Forwards token to a security component placed in front of SP [user disagreement] Doesn't agree on identity data transfer to SP Access denied [invalid certificate] Access denied Lombardy Citizen Services National Card Lombardy IdPC Lombardy Region Service Provider SIAGE Proxy re-encryption Module

A.1.2.2 1b. Citizen performs authentication via eIDAS adapter

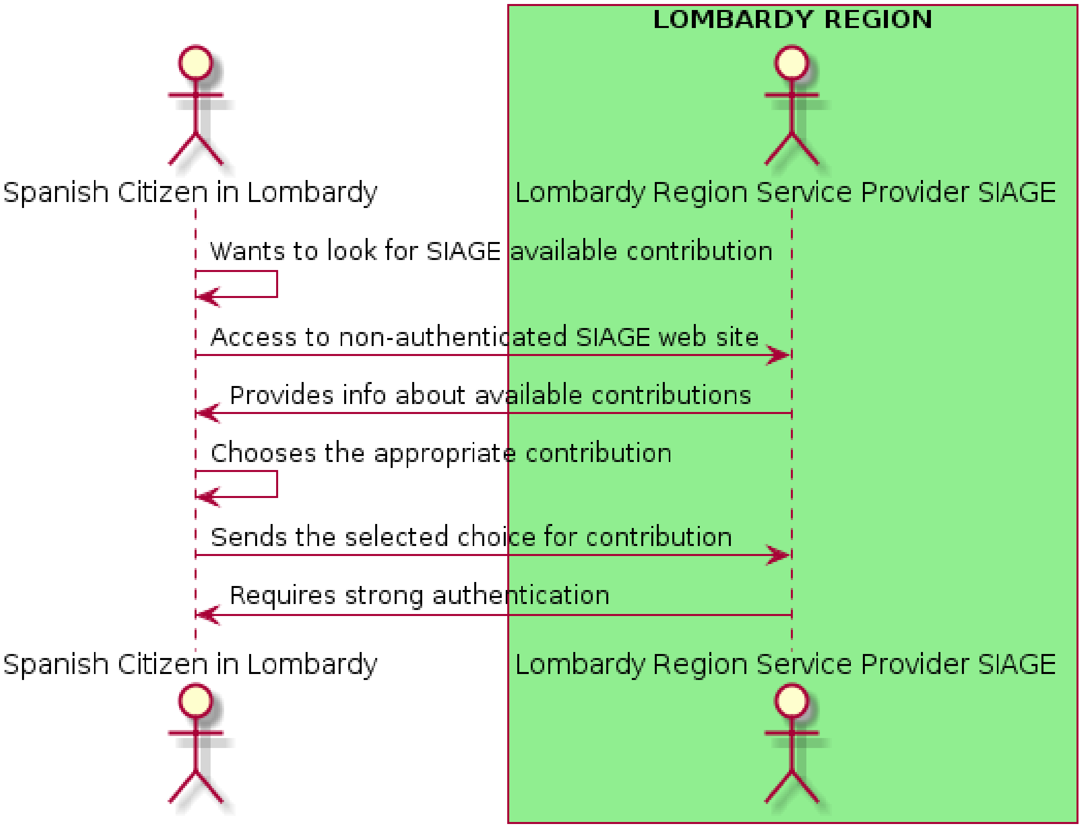
Use Case Name	**1b. Citizen performs authentication via eIDAS adapter**
ID	E.BUS-LUC-1546
Main Actor	- Citizen
Secondary Actors	- Lombardy Region IdP (IdPC) - CNS - eIDAS adapter - CREDENTIAL proxy re-encryption Module - Lombardy Region Service Provider
Pre-conditions	- User has previously inserted his CNS in smartcard reader
Post-conditions	- User is successfully authenticated into IdPC
Description	- User has inserted CNS into the smartcard reader and digits the 5-numbers associated PIN. The user certificate is examined by IdPC in order to determine if it is not revoked and not suspended. Proxy re-encryption is made. eIDAS adapter is then engaged to manage SAML response.
Image	 LOMBARDY REGION CREDENTIAL WALLET Lombardy Citizen Services National Card Lombardy IdPC Lombardy Region Service Provider SIAGE eIDAS adapter Proxy re-encryption Modul Requests client authentication through the browser Asks to insert 5-numbers smartcard associated PIN Via local components PKCS#11 or CSP Inserts 5-digits PIN Reads user's certificate from SSL session Analyses user's certificate IdPC checks if user certificate is issued by a trusted CA and if it is valid (not revoked, not suspended, not expired) this involves Certificate Revocation List check alt [valid certificate] Certificate valid Gathers user identity information Asks user if all identity data shall be transferred to Service Provider alt [user agreement] Agrees on identity data transfer to SP User token? Request Proxy re-encryption keys Encrypts data using CREDENTIAL Prepares a SAML 2.0 assertion Forwards token to a eIDAS adapter Manages any protocol differences This is a security component which consume assertion token Forward a classic SAML 2.0 assertion to a security component placed in front of SP [user disagreement] Doesn't agree on identity data transfer to SP Access denied [invalid certificate] Access denied Lombardy Citizen Services National Card Lombardy IdPC Lombardy Region Service Provider SIAGE eIDAS adapter Proxy re-encryption Modul

A.1.2.3 2. Citizen successfully access to SP

Use Case Name	**2. Citizen successfully access to SP**
ID	E.BUS-LUC-259
Main Actor	- Citizen
Secondary Actors	- Lombardy Region Service Provider - CREDENTIAL proxy re-encryption Module
Pre-conditions	- User has been successfully authenticated by IdP
Post-conditions	- User successfully access to SP
Description	- IdPC releases a SAML 2.0 assertion - user data is crypted - to SP. SP decrypts data needed.
Image	 LOMBARDY REGION CREDENTIAL WALLET Lombardy Citizen Lombardy Region Service Provider SIAGE Proxy re-encryption Module It receives the assertion from the security component Receives a SAML 2.0 assertion Decrypts needed data using CREDENTIAL Has user data in plain form Provides access User is logged into the Service Provider Lombardy Citizen Lombardy Region Service Provider SIAGE Proxy re-encryption Module

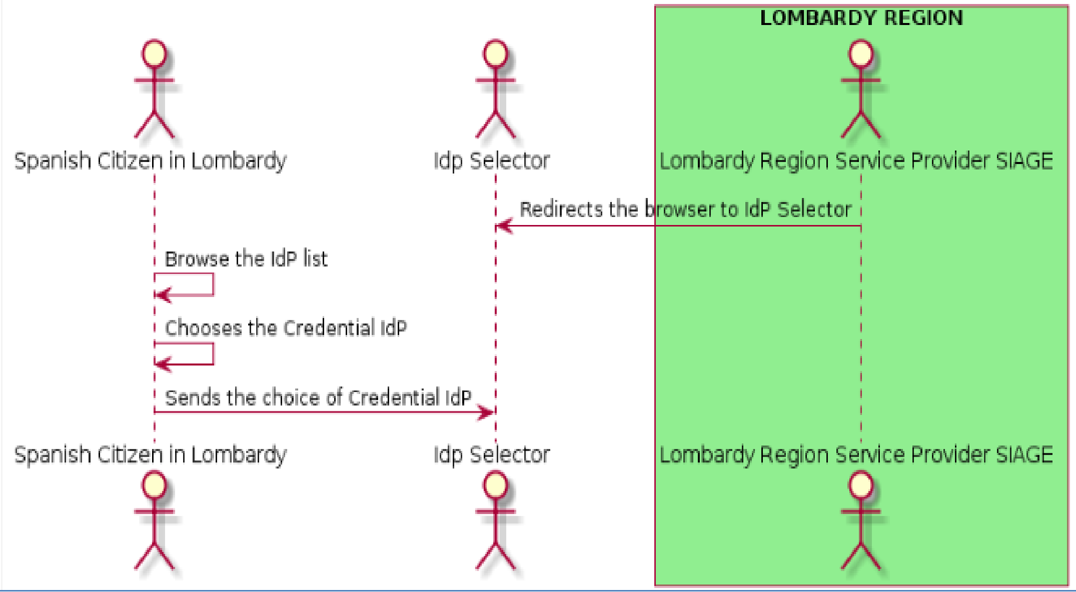
A.1.3 Foreign citizen looks for a contribution from Lombardy Region

A.1.3.1 Citizen surfs in SIAGE website

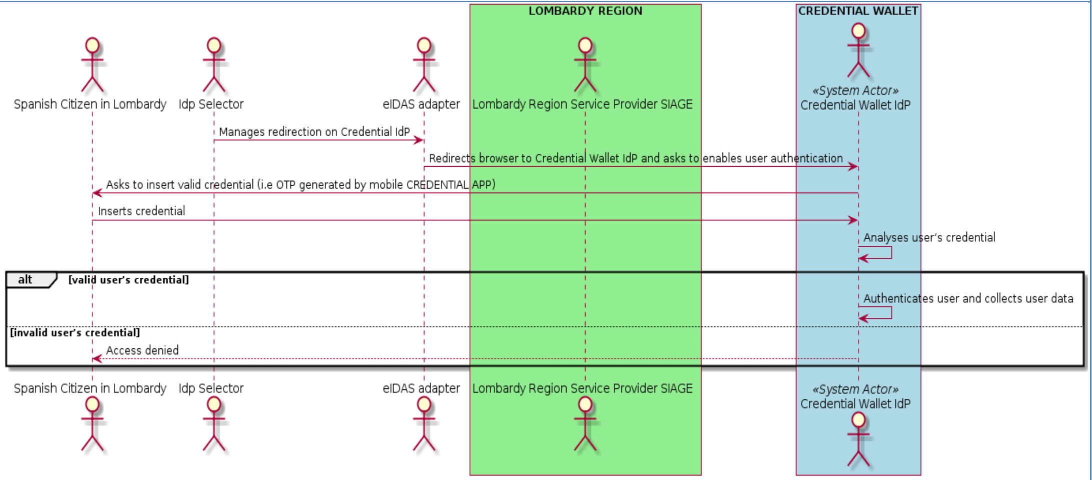
Use Case Name	Citizen surfs in SIAGE website
ID	E.BUS-LUC-274
Main Actor	- Citizen
Secondary Actors	- Lombardy Region Service Provider
Pre-conditions	- User has a needs/suitable with SIAGE contribution offer
Post-conditions	- User finds an appropriate contribution on SIAGE web site
Description	- User access to a non-authenticated area of SIAGE website in order to search for an appropriate contribution for his needs.
Image	 LOMBARDY REGION Spanish Citizen in Lombardy Lombardy Region Service Provider SIAGE Wants to look for SIAGE available contribution Access to non-authenticated SIAGE web site Provides info about available contributions Chooses the appropriate contribution Sends the selected choice for contribution Requires strong authentication Spanish Citizen in Lombardy Lombardy Region Service Provider SIAGE

A.1.4 BUC Citizen gains on-line authentication

A.1.4.1 Citizen is challenged to choose from IdP list

Use Case Name	**Citizen is challenged to choose from IdP list**
ID	E.BUS-LUC-285
Main Actor	- Citizen
Secondary Actors	- IdP selector/adapter - Lombardy Region Service Provider
Pre-conditions	- User has previously chosen a contribution request on SIAGE website which requires authentication
Post-conditions	- Spanish IdP is chosen for online authentication
Description	- User wants to use his digital identity released by a third party IdP. So he chooses the CREDENTIAL IdP from the IdP selector.
Image	 LOMBARDY REGION Spanish Citizen in Lombardy Idp Selector Lombardy Region Service Provider SIAGE Redirects the browser to IdP Selector Browse the IdP list Chooses the Credential IdP Sends the choice of Credential IdP Spanish Citizen in Lombardy Idp Selector Lombardy Region Service Provider SIAGE

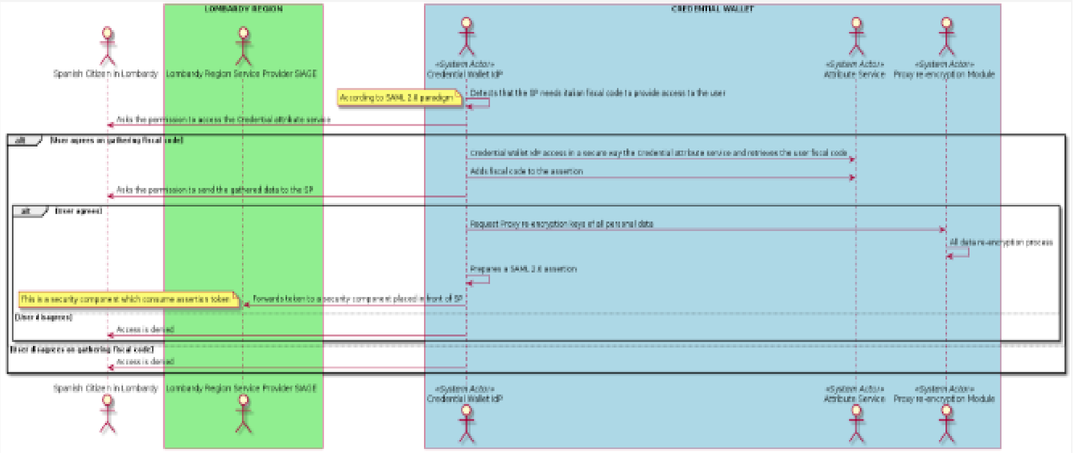
A.1.4.2 Citizen performs authentication in CREDENTIAL Wallet **IdP**

Use Case Name	**Citizen performs authentication in CREDENTIAL Wallet IdP**
ID	E.BUS-LUC-286
Main Actor	- Citizen
Secondary Actors	- CREDENTIAL Identity Provider - IdP Selector - eIDAS adapter - Lombardy Region Service Provider
Pre-conditions	- CREDENTIAL IdP is the user choice for online authentication
Post-conditions	- User gets authentication from CREDENTIAL IdP
Descriptio n	- CREDENTIAL IdP challenges Spanish citizen to use the needed credentials (i.e., OTP generated by mobile CREDENTIAL APP). User is identified and his data are collected by IdP.
Image	

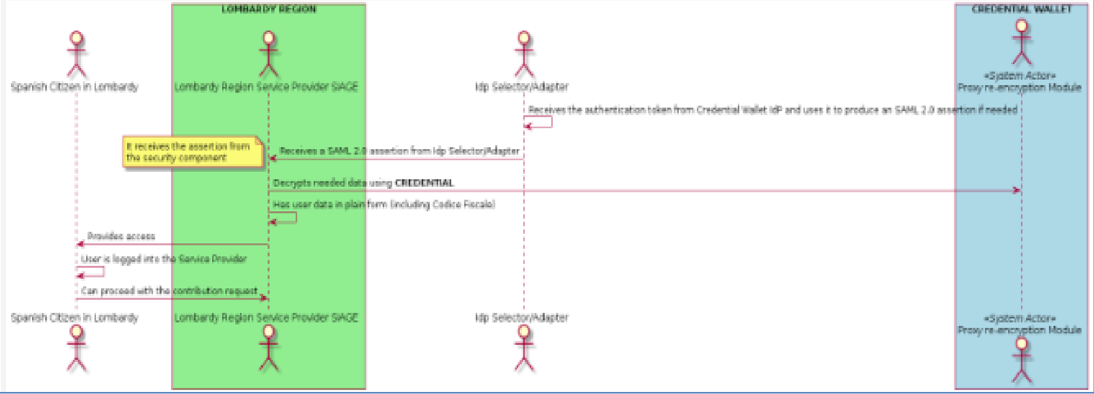
LOMBARDY REGION
CREDENTIAL WALLET
Spanish Citizen in Lombardy
Idp Selector
eIDAS adapter
Lombardy Region Service Provider SIAGE
«System Actor»
Credential Wallet IdP
Manages redirection on Credential IdP
Redirects browser to Credential Wallet IdP and asks to enables user authentication
Asks to insert valid credential (i.e OTP generated by mobile CREDENTIAL APP)
Inserts credential
Analyses user's credential
alt
[valid user's credential]
Authenticates user and collects user data
[invalid user's credential]
Access denied

A.1.5 BUC Citizen completes online request to SIAGE

A.1.5.1 Citizen adds data transferred to SP

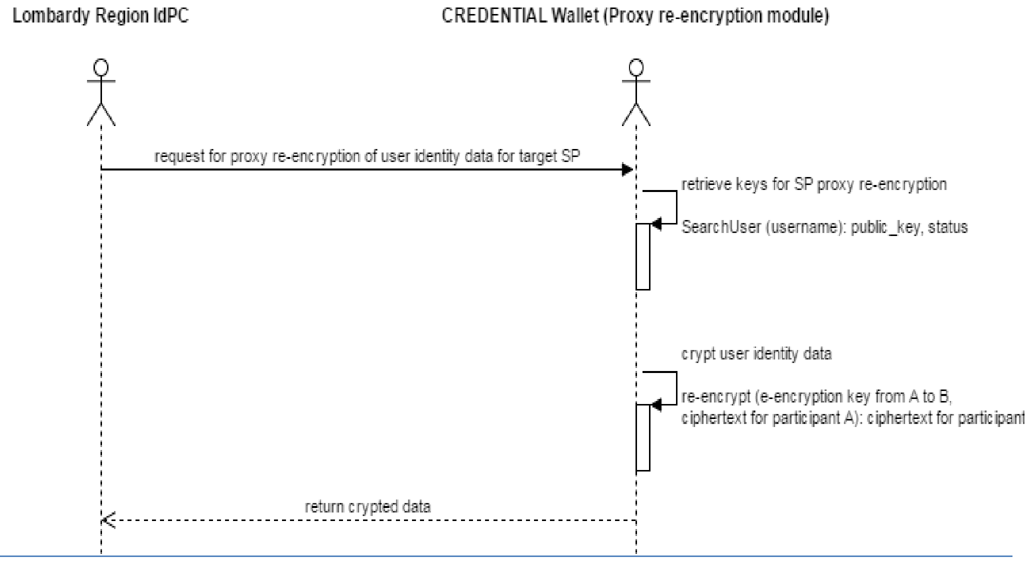
Use Case Name	Citizen adds data transferred to SP
ID	E.BUS-LUC-288
Main Actor	- Citizen
Secondary Actors	- CREDENTIAL proxy re-encryption Module - CREDENTIAL Identity Provider - Lombardy Region Service Provider - CREDENTIAL Attribute Service
Pre-conditions	- User has been successfully authenticated by IdP
Post-conditions	- Spanish IdP releases SAML 2.0 authentication enriched with user data retrieved from Wallet
Description	- SIAGE needs one more data from user: his “Codice Fiscale” (Italian tax unique id). This data has been previously stored in CREDENTIAL Wallet, so citizen can collect it and add this domain-specific data in order to transmit them to the SP. Proxy re-encryption is made.
Image	 LOMBARDY REGION CREDENTIAL WALLET Spanish Citizen in Lombardy Lombardy Region Service Provider SIAGE «System Actor» Credential Wallet IdP «System Actor» Attribute Service «System Actor» Proxy re-encryption Module According to SAML 2.0 paradigm Detects that the SP needs italian fiscal code to provide access to the user Asks the permission to access the Credential attribute service alt [User agrees on gathering fiscal code] Credential wallet idP access in a secure way the credential attribute service and retrieves the user fiscal code Adds fiscal code to the assertion Asks the permission to send the gathered data to the SP alt [User agrees] Request Proxy re-encryption keys of all personal data All data re-encryption process Prepares a SAML 2.0 assertion This is a security component which consume assertion token Forwards token to a security component placed in front of SP [User disagrees] Access is denied [User disagrees on gathering fiscal code] Access is denied

A.1.5.2 Citizen access to SIAGE and fills request

Use Case Name	Citizen access to SIAGE and fills request
ID	E.BUS-LUC-289
Main Actor	- Citizen
Secondary Actors	- Lombardy Region Service Provider - CREDENTIAL proxy re-encryption Module - IdP selector/adapter
Pre-conditions	- CREDENTIAL IdP has released an identity assertion
Post-conditions	- User is logged into SP
Description	- All data needed to SIAGE are now ready. SP receives those data in a crypted form and, after a successful decryption, can use them to let user proceeds with the contribution request.
Image	 LOMBARDY REGION CREDENTIAL WALLET Spanish Citizen in Lombardy Lombardy Region Service Provider SIAGE Idp Selector/Adapter «System Actor» Proxy re-encryption Module Receives the authentication token from Credential Wallet IdP and uses it to produce an SAML 2.0 assertion if needed It receives the assertion from the security component Receives a SAML 2.0 assertion from Idp Selector/Adapter Decrypts needed data using CREDENTIAL Has user data in plain form (including Codice Fiscale) Provides access User is logged into the Service Provider Can proceed with the contribution request Spanish Citizen in Lombardy Lombardy Region Service Provider SIAGE Idp Selector/Adapter «System Actor» Proxy re-encryption Module

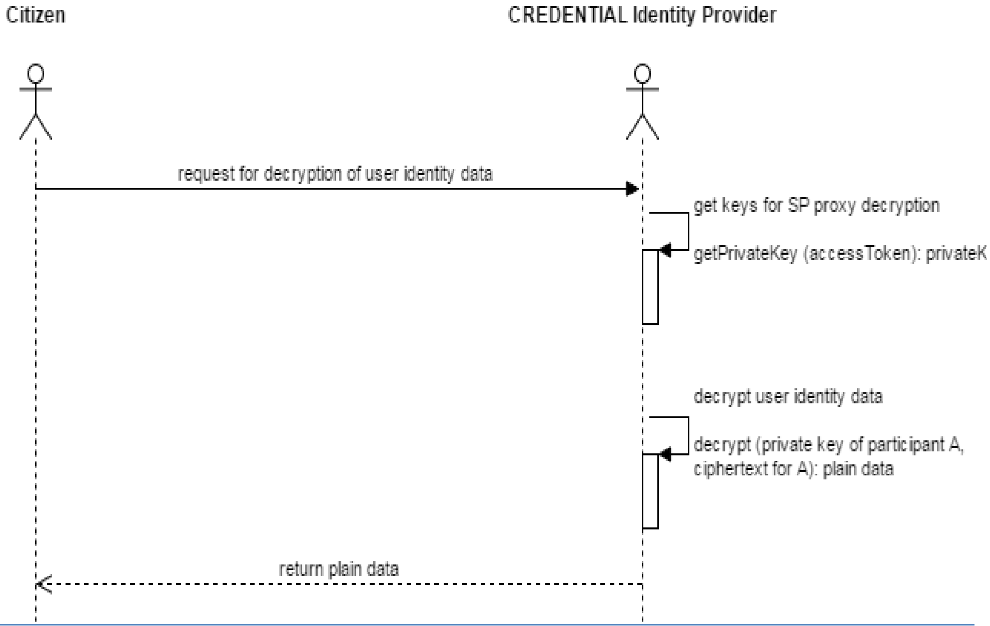
A.2 Technical Use Cases

A.2.1 Proxy re-encryption of user data

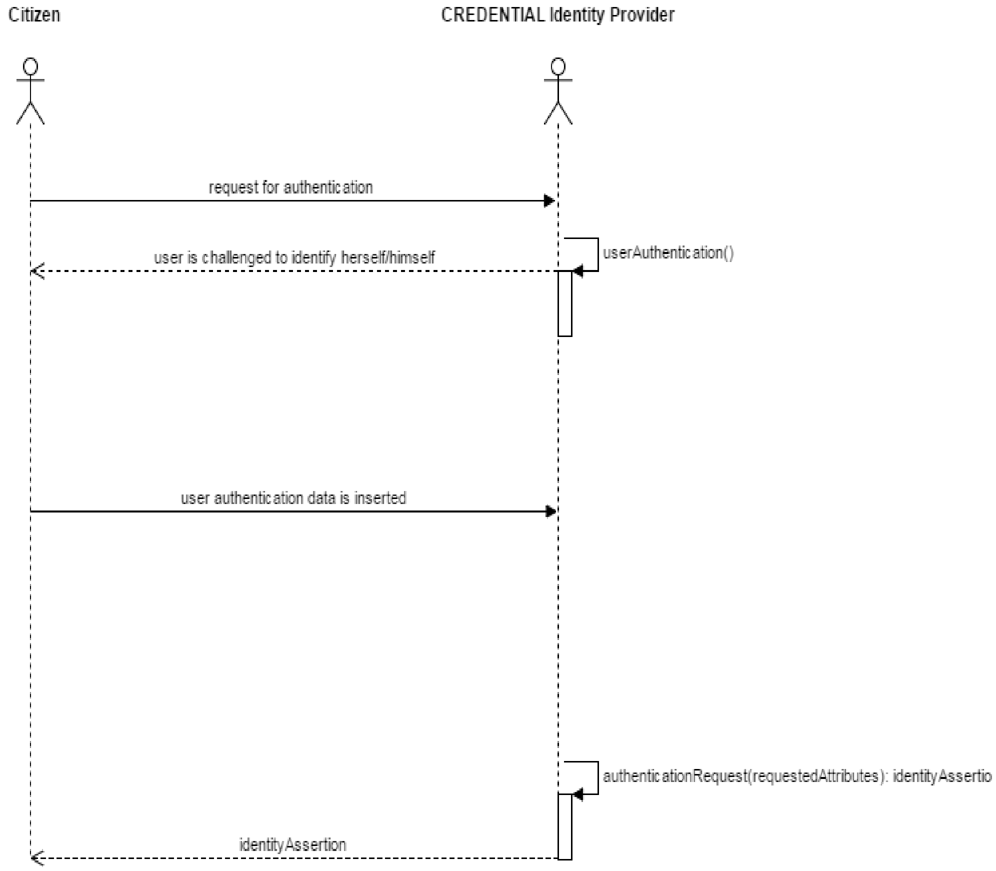
Use Case Name	Proxy re-encryption of user data
ID	EGOV-TUC-001
Main Actor	- Lombardy Region IdPC
Secondary Actors	- CREDENTIAL proxy re-encryption Module
Pre-conditions	- User data is stored into wallet (in a crypted form)
Post-conditions	- User data is re-encrypted for target SP
Description	- IdPC access to CREDENTIAL platform to proxy re-encrypt user data. CREDENTIAL proxy re-encryption Module is part of “Data Management Service” located in the Data Service layer of CREDENTIAL platform.
Image	

Lombardy Region IdPC
CREDENTIAL Wallet (Proxy re-encryption module)
request for proxy re-encryption of user identity data for target SP
retrieve keys for SP proxy re-encryption
SearchUser (username): public_key, status
crypt user identity data
re-encrypt (e-encryption key from A to B, ciphertext for participant A): ciphertext for participant
return crypted data

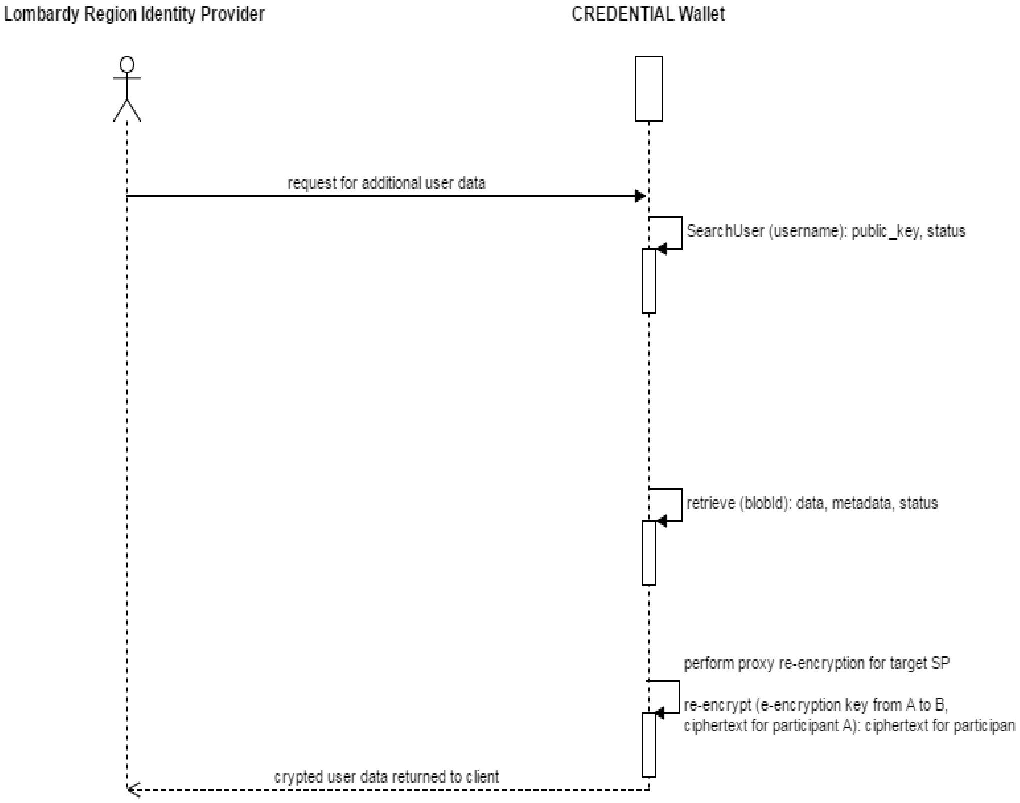
A.2.2 User data decryption

Use Case Name	User data decryption
ID	EGOV-TUC-002
Main Actor	- Lombardy Region Service Provider
Secondary Actors	- CREDENTIAL proxy re-encryption Module
Pre-conditions	- User data is transmitted in a crypted form to SP
Post-conditions	- SP is able to manage plain (decrypted) user data
Description	- Shibboleth SP access to CREDENTIAL platform to decrypt data. CREDENTIAL proxy re-encryption Module is part of "Data Management Service" located in the Data Service layer of CREDENTIAL platform.
Image	 Citizen CREDENTIAL Identity Provider request for decryption of user identity data get keys for SP proxy decryption getPrivateKey (accessToken): privateK decrypt user identity data decrypt (private key of participant A, ciphertext for A): plain data return plain data

A.2.3 CREDENTIAL platform used as IdP

Use Case Name	CREDENTIAL platform used as IdP
ID	EGOV-TUC-003
Main Actor	- Citizen
Secondary Actors	- CREDENTIAL Identity Provider
Pre-conditions	- User needs authentication to access to a protected area of a SP
Post-conditions	- User is authenticated in CREDENTIAL IdP
Description	- CREDENTIAL platform plays a role of and IdP and authenticates user. CREDENTIAL Identity Provider is part of "Authentication Service" located in the Business layer of CREDENTIAL platform.
Image	 Citizen CREDENTIAL Identity Provider request for authentication user is challenged to identify herself/himself userAuthentication() user authentication data is inserted authenticationRequest(requestedAttributes): identityAssertio identityAssertion

A.2.4 CREDENTIAL Wallet access to retrieve user data

Use Case Name	CREDENTIAL Wallet access to retrieve user data
ID	EGOV-TUC-004
Main Actor	- Lombardy Region Identity Provider
Secondary Actors	- CREDENTIAL Wallet
Pre-conditions	- SP needs some user data not included in native IdP assertion
Post-conditions	- IdP delivers an assertion which includes user identity data and additional user data
Description	- CREDENTIAL platform is used here as a secure store of additional user data. The component used is the "Data Store", located in the Data Access layer of CREDENTIAL platform.
Image	 Lombardy Region Identity Provider CREDENTIAL Wallet request for additional user data SearchUser (username): public_key, status retrieve (blobId): data, metadata, status perform proxy re-encryption for target SP re-encrypt (e-encryption key from A to B, ciphertext for participant A): ciphertext for participan crypted user data returned to client

B eHealth Use Cases

This section defines messages, terminologies and value sets for implementing the eHealth pilot. For doing so, a model driven approach is used which directly connects to the CREDENTIAL business use case definitions. These business level viewpoints are decomposed and refined into logical and technical specifications of the underlying IT services:

- Section B.1 specifies the logical use cases, considering functionalities as well as information models.
- Section B.2 specifies profiles on the IHE profiles which are used in the eHealth pilot. For each IHE transaction used, the specific constraints and configurations for CREDENTIAL are given.
- Section B.3 further refines the logical use cases into technical use cases that determine the interplay of PHR services and CREDENTIAL services.

The figure below shows where to find the technical specifications for the aspects of the eHealth pilot that cover PHR interaction, CREDENTIAL security, and the integration of both.

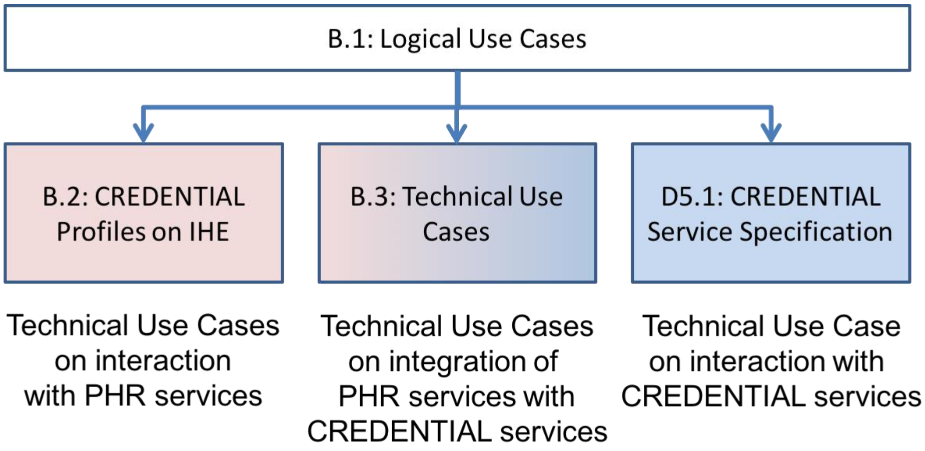

B.1: Logical Use Cases
B.2: CREDENTIAL Profiles on IHE
B.3: Technical Use Cases
D5.1: CREDENTIAL Service Specification
Technical Use Cases on interaction with PHR services
Technical Use Cases on integration of PHR services with CREDENTIAL services
Technical Use Case on interaction with CREDENTIAL services

Figure 27: Outline of Appendix B

B.1 Logical Use Cases

This section is organized to reflect the Service Functional Model as given in Section 5.4.3. For each logical use case first a generic flow of control is given in narratives. Aspects of that flow that map onto interactions with or among technical components are then formally defined as use cases together with sequence diagrams that show these interactions in detail.

B.1.1 Logical Actors

For defining the logical use cases for the CREDENTIAL eHealth pilot the coarse-grained sketches provided in Section 5.2 will be broken down into logical actors with each actor consuming and/or providing defined functionalities through defined (logical) services.

The figure below shows the eHealth pilot specific logical actors as they will be used for defining the logical use cases.

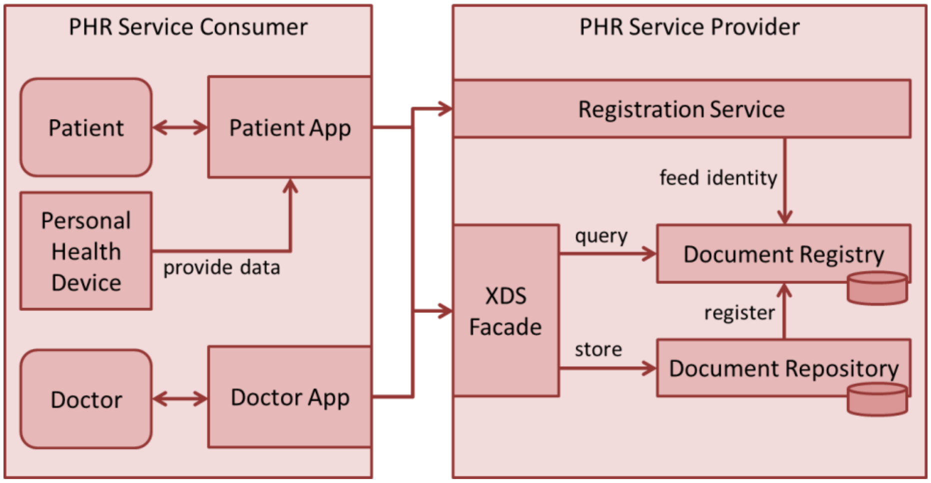


Figure 28: Logical Actors of the CREDENTIAL eHealth Pilot

On the client side, patients and doctors interact with the Personal Health Record (PHR) infrastructure through dedicated Apps. While the Patient App is considered to be deployed on a smartphone, the App for doctors will be designed for larger devices, e.g., tablets or convertibles. Patients will be equipped with personal health devices (e.g., digital scales) which connect to the Patient App through mechanisms provided by the underlying hardware and operating system.

The design of the PHR itself is strictly oriented towards the IHE Cross-Enterprise Document Sharing (XDS) profile. This profile defines a registry for metadata management and a repository for data management as core components. In order to make existing XDS implementations usable for CREDENTIAL, a façade intercepts all XDS messages for connecting with the CREDENTIAL security services. The CREDENTIAL PHR will be linked to an existing CREDENTIAL user account. For synchronizing identifiers, a Registration Service is introduced that, e.g., feeds the patient’s CREDENTIAL account identifier into the PHR registry for initializing a new health record.

The CREDENTIAL logical actors as to be used for the definition of the eHealth logical use cases are taken from CREDENTIAL D5.1 “Functional Design”. The figure below shows the respective definitions. While Participant Services are consumed through a client-side library by the CREDENTIAL Patient and Doctor App only, all services deployed within the cloud are assumed to accessible to both App-deployed services and PHR services through defined service interfaces.

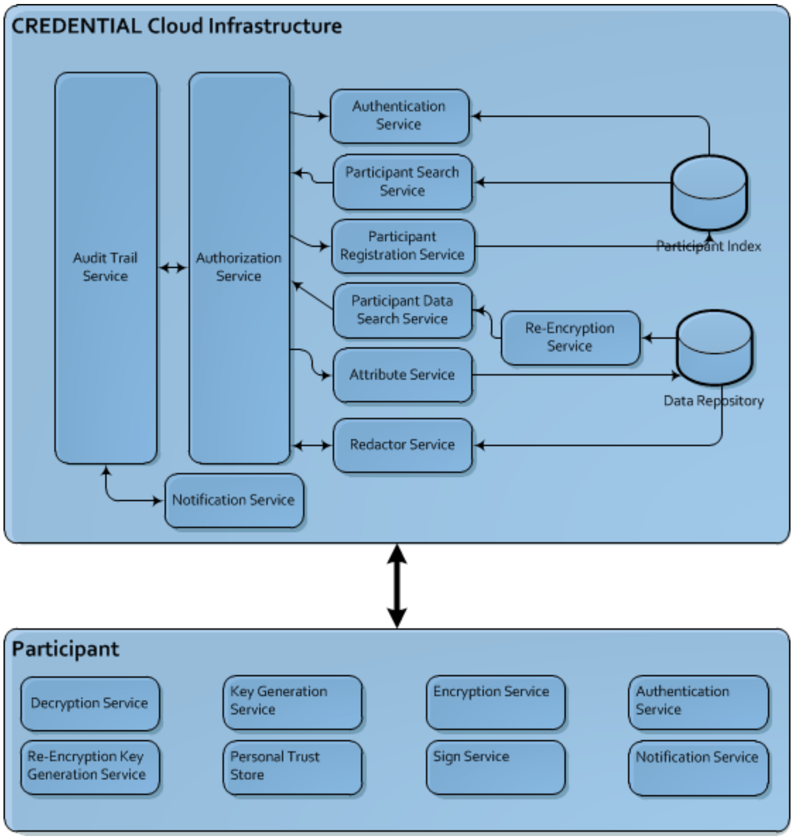


Figure 29: CREDENTIAL Logical Actors (from D5.1)

B.1.2 Logical Use Case: Initialize PHR

This logical use case initializes a PHR instance within a CREDENTIAL account. It is assumed that there may only be a single PHR instance within each CREDENTIAL account (which is also a restriction imposed by the IHE XDS integration profile which relies on a 1:1 correspondence between a patient identifier and a health record).

For reasons of simplicity it is further assumed that only the patient himself is allowed to set up a PHR within his own CREDENTIAL account. This restriction allows the CREDENTIAL Wallet to validate the permissions of the requestor without any need for further interacting with any other services.

The logical use case "Initialize PHR" not only registers the patient with CREDENTIAL and the PHR but even establishes a trust relationship between the CREDENTIAL Patient App on the patient's smartphone and the PHR services. This is done through sharing a TLS certificate between the PHR Registration Service and the Patent App. This certificate is used in conjunction with the IHE ATNA connection establishment which requires a mutual TLS authentication between the nodes that share data through IHE XDS.

The generic flow of control is as follows:

#	Flow of Control and Data	Definition

1	The doctor informs the patient about CREDENTIAL. The patient signs a consent form which is kept by the doctor. The doctor hands over to the patient the Personal Health Devices for monitoring the patient's treatment.	
2	The doctor registers the patient as a CREDENTIAL eHealth pilot participant with the PHR Registration Service.	B.1.2.1: Register Patient as eHealth Pilot Participant
3	The patient installs the CREDENTIAL Patient App on his smartphone and activates the registration. Through the Patient App the patient's CREDENTIAL account is linked with the patient's PHR. A certificate for IHE ATNA secure communication is issued and shared.	0: Link PHR to CREDENTIAL Account
4	The Patient App generates a PHR-Key for the patient's PHR and stores the key to the CREDENTIAL Data Repository.	B.1.2.3: Create and Register PHR-Key
5	The patient registers the doctor as his diabetologist (or family doctor).	See Logical Use Case: Set User Role
6	The doctor uploads a scan of the patient's signed consent document to the PHR.	See Logical Use Case: Provide Documents
7	The patient pairs his Personal Health Devices with his smartphone and authorizes the CREDENTIAL Patient App for accepting data from these devices.	Vendor specific interaction among devices; left open to the implementation.

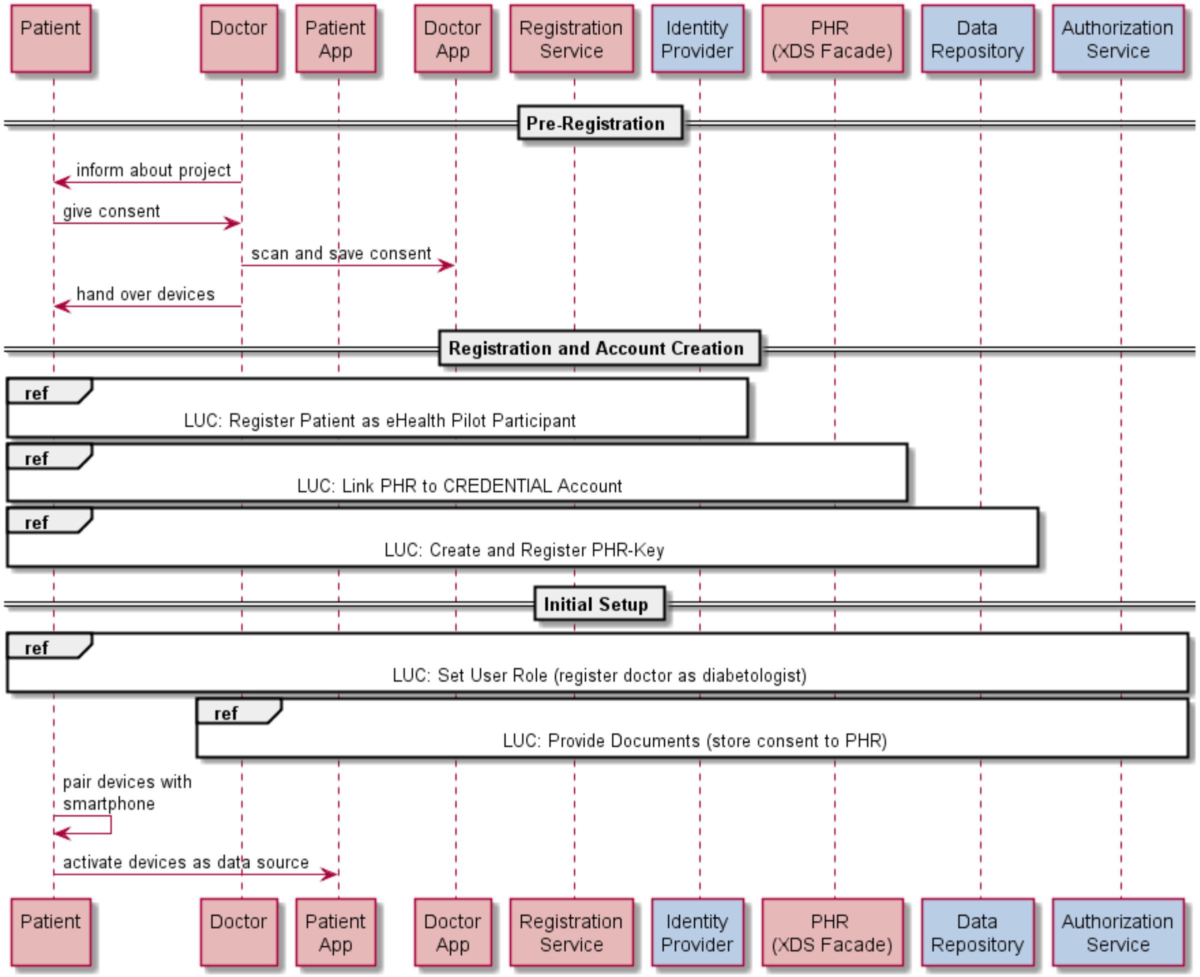
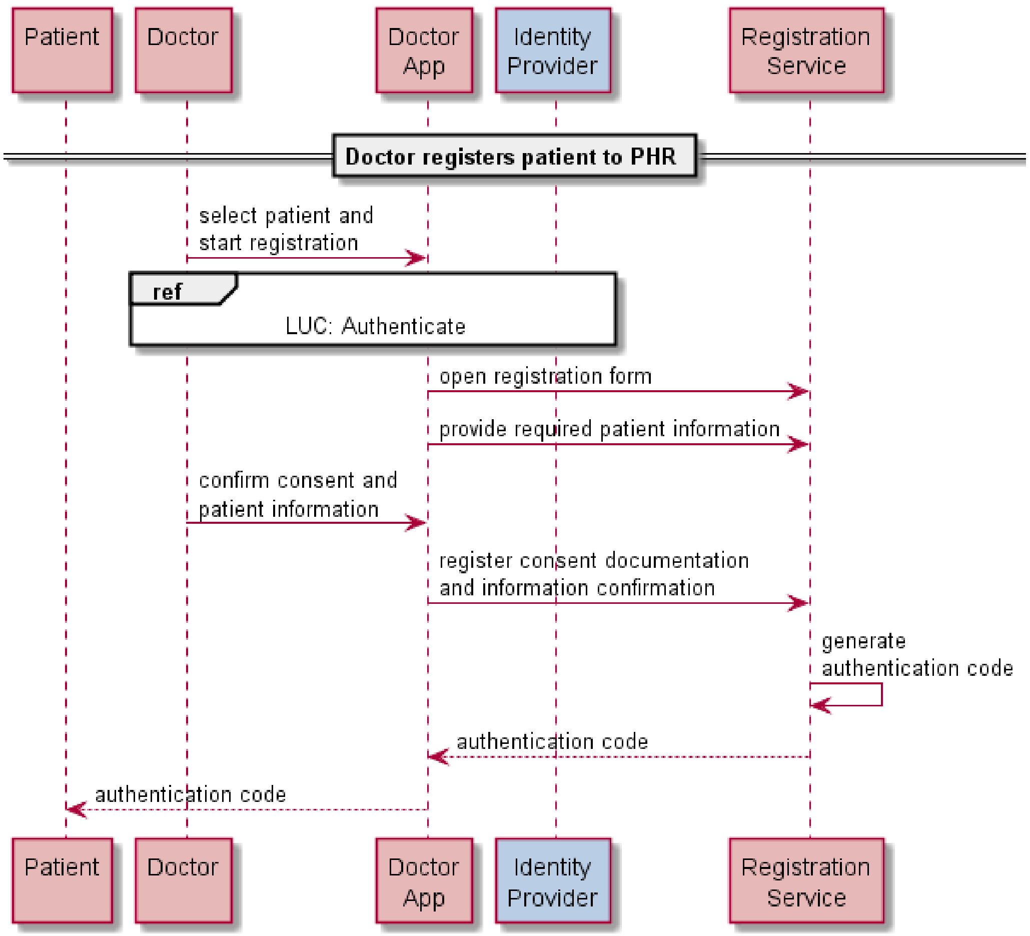


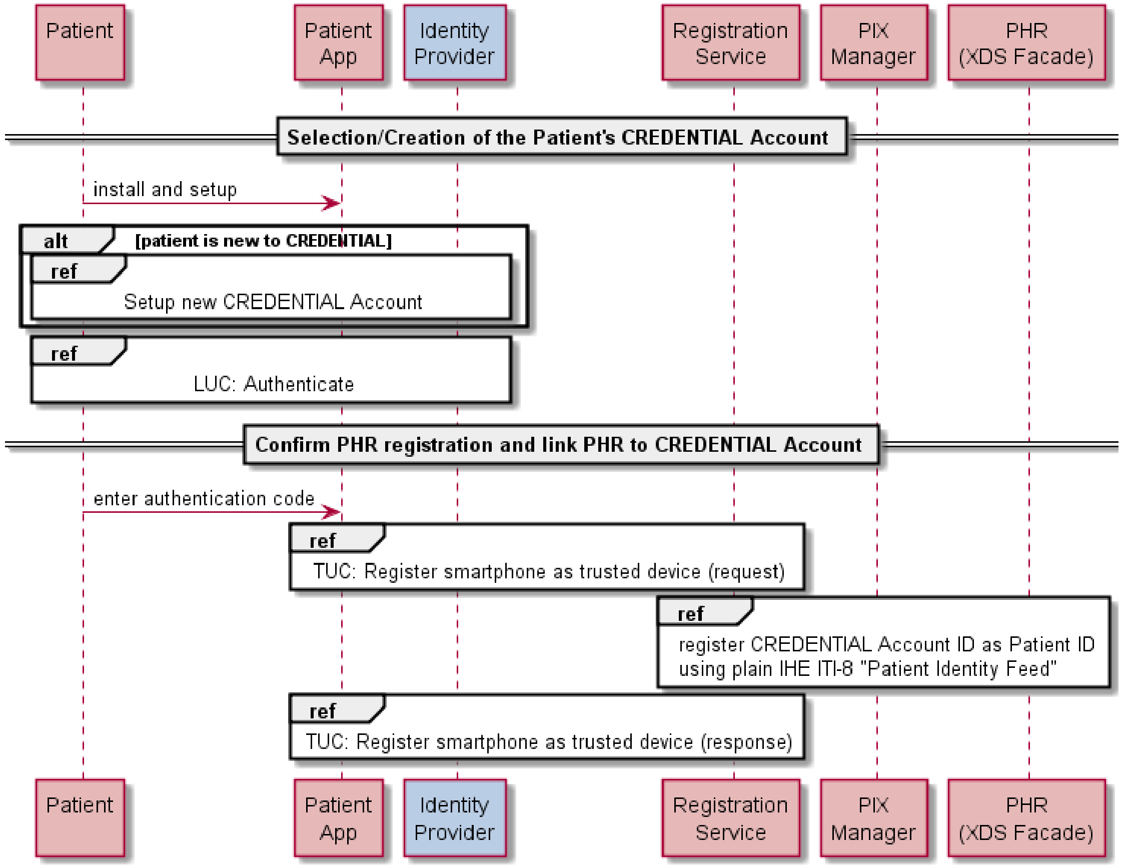
Figure 30: LUC010 Initialize PHR

Note that in this figure, references only span the message that triggers the included use cases.

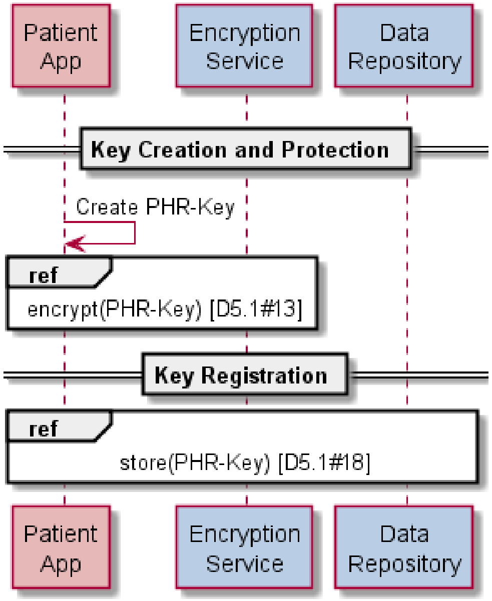
B.1.2.1 Register Patient as Pilot Participant

Use Case Name	Register Patient as eHealth Pilot Participant
ID	E.HLT-LUC-011
Main Actor	- Doctor
Secondary Actors	- CREDENTIAL Doctor App - Registration Service - Identity Provider - Patient
Pre-conditions	- Patient has given consent to the doctor - Doctor is registered as CREDENTIAL participant
Post-conditions	- PHR is set up - Patient is prepared for linking his CREDENTIAL account with the PHR
Description	The doctor registers the patient as a CREDENTIAL eHealth pilot participant with the PHR provider.
Image	 Patient Doctor Doctor App Identity Provider Registration Service Doctor registers patient to PHR select patient and start registration ref LUC: Authenticate open registration form provide required patient information confirm consent and patient information register consent documentation and information confirmation generate authentication code authentication code authentication code Patient Doctor Doctor App Identity Provider Registration Service

B.1.2.2 Link PHR to CREDENTIAL Account

Use Case Name	Link PHR to CREDNETIAL Account
ID	E.HLT-LUC-012
Main Actor	- Patient
Secondary Actors	- CREDENTIAL Patient App - Registration Service - Identity Provider - PIX Manager - Document Registry (via XDS Façade)
Pre-conditions	- PHR has been initialized for the patient - Patient received the authentication code for one-time-access to his PHR
Post-conditions	- PHR is linked with an CREDENTIAL account - Patient App is installed and ready for use - Patient's smartphone is prepared to act as a trusted device with respect to the conditions of the IHE ATNA integration profile.
Description	The patient installs the CREDENTIAL Patient App on his smartphone and activates the registration. If needed, a new CREDENTIAL account is setup. The patient's CREDENTIAL account is linked with the patient's PHR. The patient's smartphone receives a certificate to establish an IHE ATNA secured communication with the PHR.
Image	 Patient Patient App Identity Provider Registration Service PIX Manager PHR (XDS Facade) Selection/Creation of the Patient's CREDENTIAL Account install and setup alt [patient is new to CREDENTIAL] ref Setup new CREDENTIAL Account ref LUC: Authenticate Confirm PHR registration and link PHR to CREDENTIAL Account enter authentication code ref TUC: Register smartphone as trusted device (request) ref register CREDENTIAL Account ID as Patient ID using plain IHE ITI-8 "Patient Identity Feed" ref TUC: Register smartphone as trusted device (response) Patient Patient App Identity Provider Registration Service PIX Manager PHR (XDS Facade)

B.1.2.3 Create and Register PHR-Key

Use Case Name	Create and Register PHR-Key
ID	E.HLT-LUC-013
Main Actor	- CREDENTIAL Patient App
Secondary Actors	- Encryption Service - Data Repository
Pre-conditions	- Patient has the Patient App installed on his smartphone - Patient is registered as CREDENTIAL participant - Patient has successfully been registered as a PHR participant - Patient is logged in to CREDENTIAL
Post-conditions	- PHR-Key is generated and stored to the CREDENTIAL Wallet
Description	The Patient App generates a PHR-Key for the patient's PHR and stores the key to the CREDENTIAL Data Repository. Authorized users may get access to the PHR-Key through CREDENTIAL's proxy re-encryption service.
Image	 Patient App Encryption Service Data Repository Key Creation and Protection Create PHR-Key ref encrypt(PHR-Key) [D5.1#13] Key Registration ref store(PHR-Key) [D5.1#18] Patient App Encryption Service Data Repository

B.1.3 Logical Use Case: Provide Documents

This logical use case implements the upload of a set of medical documents to a patient's Personal Health Record (PHR). For this use case three scenarios must be considered:

- Scenario 1: A document is to be uploaded by the patient. The data is available as a document on the patient's smartphone (e.g., rendered from an interactive form by the Patient App). The PHR-Key is available on the patient's smartphone.
- Scenario 2: The data is to be uploaded by a doctor. The data is available as a document on the doctor's tablet (e.g., rendered from an interactive form by the Doctor App). The doctor is granted a role by the patient that permits him to upload data to the patient's PHR.

- Scenario 3: Monitoring data captured by a patient's Personal Health Devices is to be uploaded. The data has been transmitted to the Patient App from the Personal Health Device via Bluetooth. The PHR-Key is available on the patient's smartphone.

The generic flow of control is as follows:

#	Flow of Control and Data	Definition
1	The user (patient or doctor) logs in to CREDENTIAL through his App by authenticating with the CREDENTIAL Identity Provider.	See Logical Use Case: Authentication
2	Scenario 2 only: The user obtains the PHR-Key for the patient's PHR	See Logical Use Case: Obtain PHR-Key
3	Scenario 3 only: The Patient App on the patient's smartphone renders the captured device data as a document.	The rendering of a document from device data is left to the implementation.
4	The Patient/Doctor App generates a key for each document and encrypts the document with that key. The key itself is encrypted by the PHR-key. Document and encrypted key are capsuled as a Documents Sharing Package.	B.1.3.1: Encrypt and Pack Document
5	The Patient/Doctor App sends an ITI-41 message to the XDS Façade.	See Section B.2
6	The XDS Façade validates the authenticity and authorization of the requestor.	B.1.3.2: Validate Requestor Authenticity and Authorization
7	The XDS Façade stores and registers the provided documents and processes all associations linked with these documents (e.g., deprecating an updated document)	See Section B.2
8	The XDS Façade triggers the Notification Service to notify authorized users who registered for the patient's account about the new documents	See Logical Use Case: Trigger Event
9	The XDS Façade writes an audit trail entry	See Logical Use Case: Write Audit Trail Entry

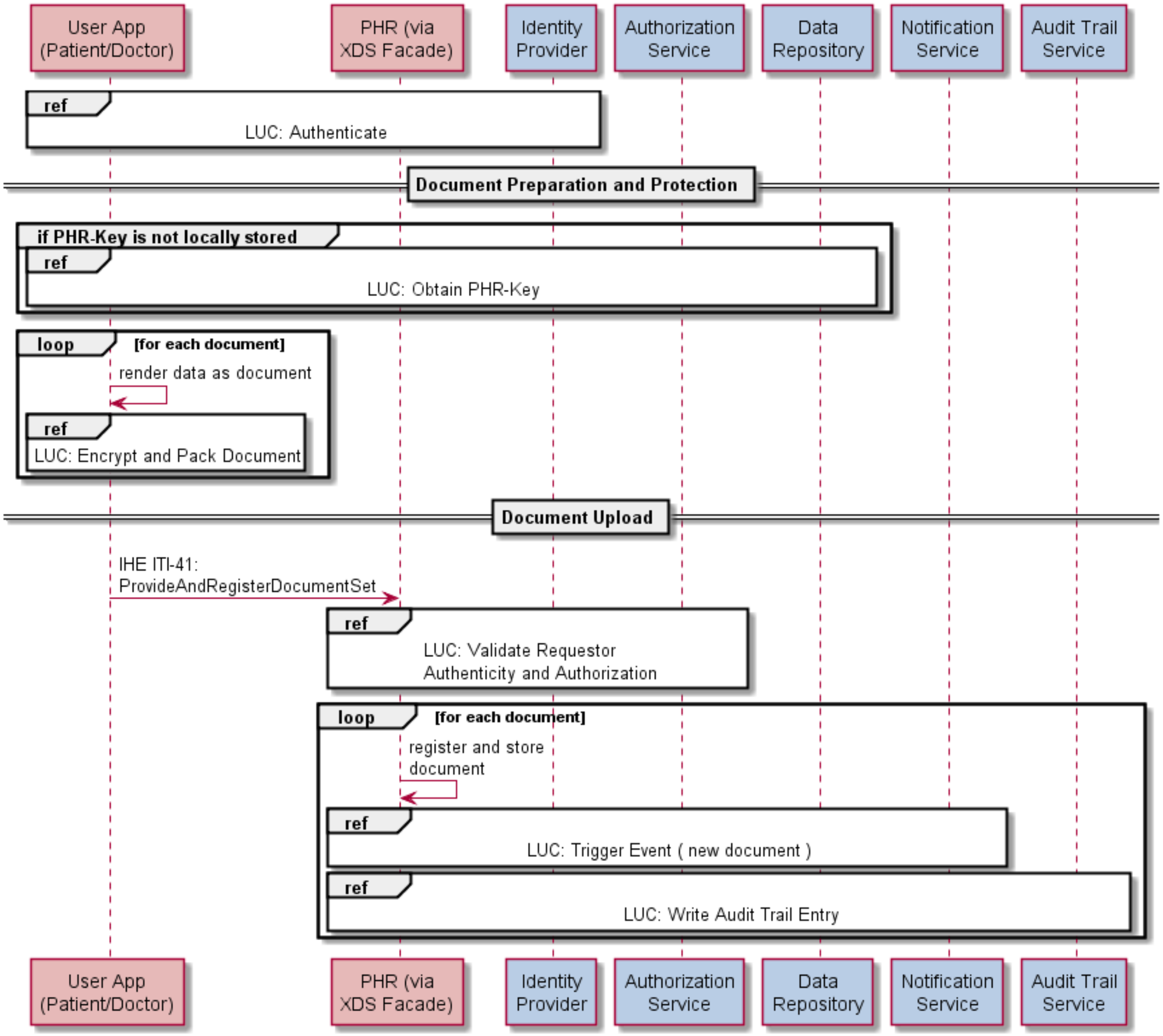
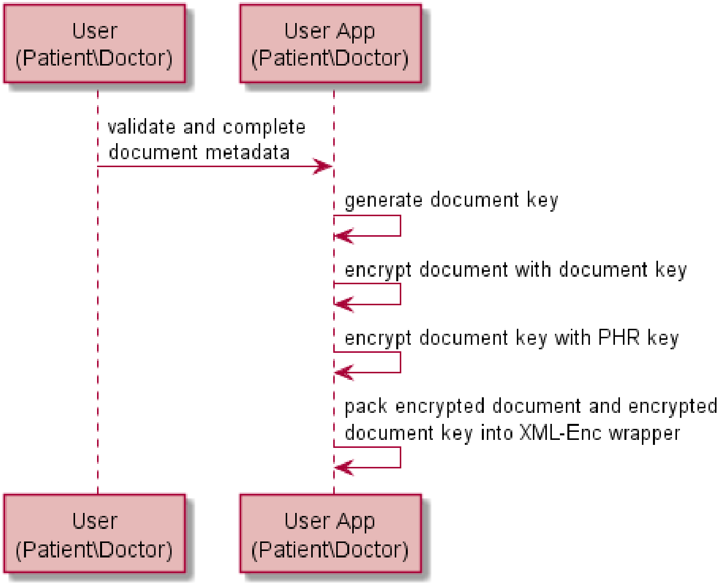


Figure 31: LUC020 Provide Documents

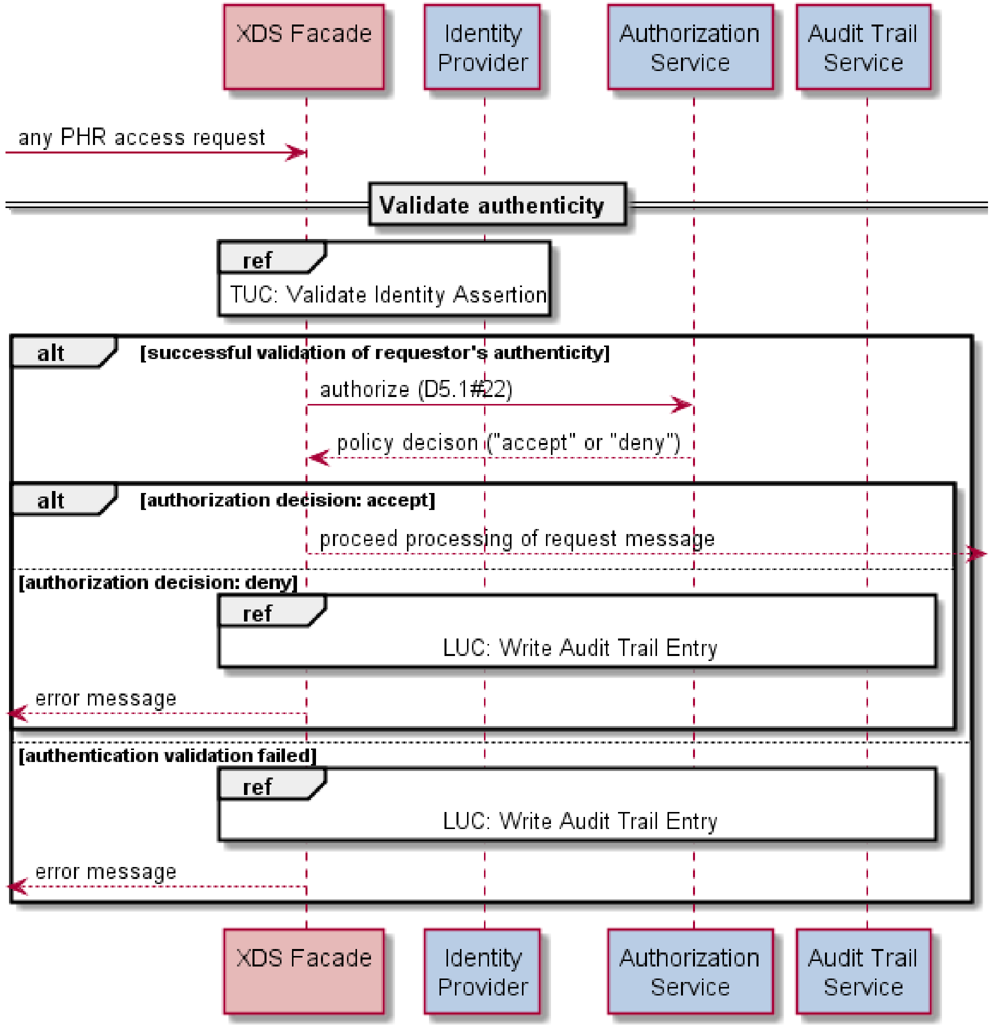
Note that in the above figure, references only span the message that triggers the included use cases.

B.1.3.1 Encrypt and Pack Document

Use Case Name	Encrypt and Pack Document
ID	E.HLT-LUC-021
Main Actor	- CREDENTIAL App (Patient App or Doctor App)
Secondary Actors	- User (patient or doctor)
Pre-conditions	- User has the CREDENTIAL eHealth App installed and activated - Documents for upload have been selected and approved by the user - PHR-Key for the target PHR is accessible to the User App
Post-conditions	- Document is safeguarded and prepared for being uploaded to the patient's PHR
Description	The Patient/Doctor App generates a key for each document and encrypts the document with that key. The key itself is encrypted by the PHR-key. Document and encrypted key are capsuled as a Documents Sharing Package.
Image	 User (Patient\Doctor) User App (Patient\Doctor) validate and complete document metadata generate document key encrypt document with document key encrypt document key with PHR key pack encrypted document and encrypted document key into XML-Enc wrapper User (Patient\Doctor) User App (Patient\Doctor)

B.1.3.2

B.1.3.3 Validate Requestor Authenticity and Authorization

Use Case Name	Validate Requestor Authenticity and Authorization
ID	E.HLT-LUC-022
Main Actor	- XDS Façade
Secondary Actors	- Authorization Service
Pre-conditions	- A PHR access requests has been accepted by the PHR Façade
Post-conditions	- The access requests is validated with respect to the permission of the requestor to access the affected PHR
Description	The XDS Façade validates the authenticity and authorization of the requestor: - Is the identity claim authentic and valid? - Has the requestor sufficient permission to perform the requested operation on the identified PHR?
Image	 XDS Facade Identity Provider Authorization Service Audit Trail Service any PHR access request Validate authenticity ref TUC: Validate Identity Assertion alt [successful validation of requestor's authenticity] authorize (D5.1#22) policy decison ("accept" or "deny") alt [authorization decision: accept] proceed processing of request message [authorization decision: deny] ref LUC: Write Audit Trail Entry error message [authentication validation failed] ref LUC: Write Audit Trail Entry error message XDS Facade Identity Provider Authorization Service Audit Trail Service

B.1.4 Logical Use Case: Query Documents

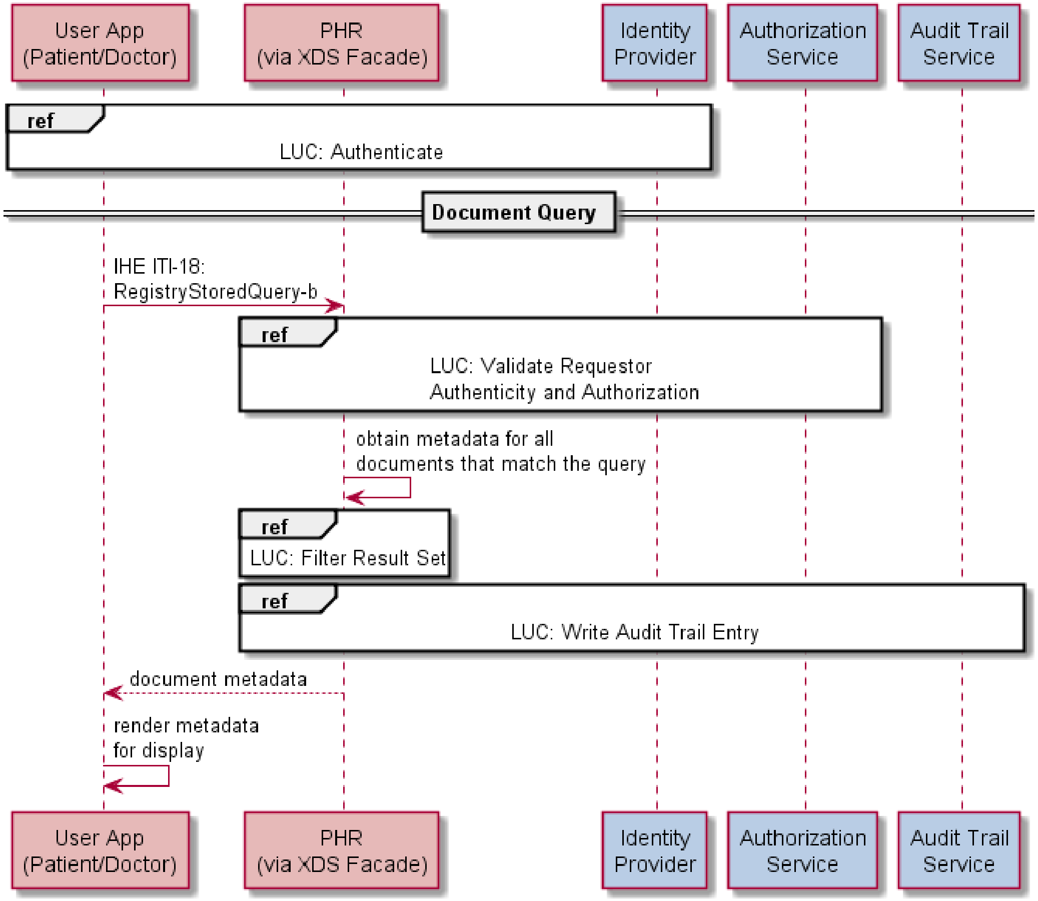
This use case provides means to discover documents within a patient's PHR that match with given query parameters (e.g., type of document):

- Find all active documents assigned with an identified account (PHR instance)
- Find all active documents assigned with an identified account (PHR instance) that cover care events within a defined time range
- Find all active documents assigned with an identified account (PHR instance) that are of a defined class/type
- Find the metadata for an identified document

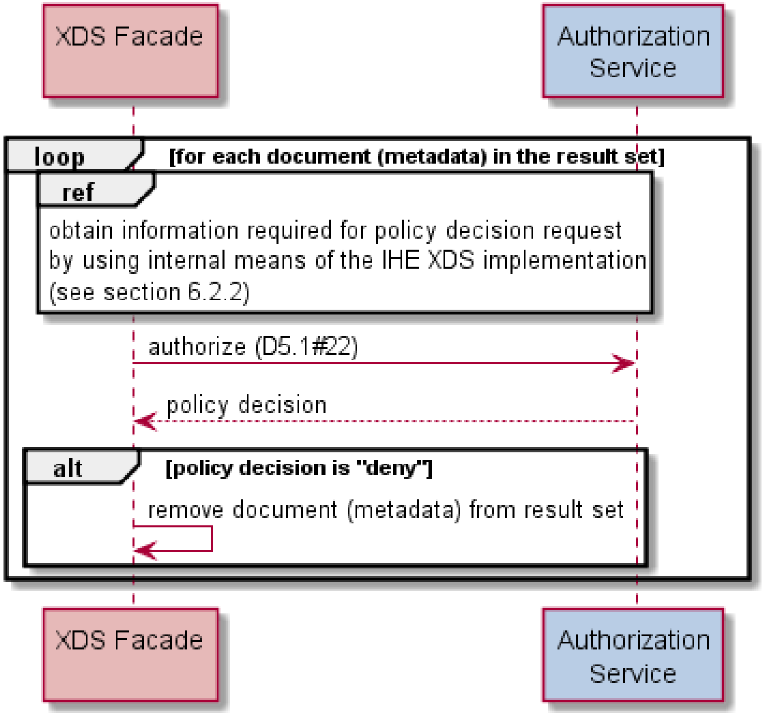
As a result, the set of metadata of all documents which match the query together with all document relationships that are defined for these documents are provided. For fetching the documents see Logical Use Case: Retrieve Documents.

The generic flow of control is as follows:

#	Flow of Control and Data	Definition
1	The user (patient or doctor) logs in to CREDENTIAL by authenticating with the CREDENTIAL Identity Provider.	See Logical Use Case: Authentication
2	The Patient/Doctor App sends an ITI-18 message to the XDS Façade.	See Section B.2
3	The XDS Façade validates the authenticity and authorization of the requestor.	B.1.3.2: Validate Requestor Authenticity and Authorization
4	The XDS Façade queries the Document Registry for all requested document metadata and associations	See Section B.2
5	The XDS Façade filters out all metadata sets on documents that the requestor is not authorized to access	B.1.4.1: Filter Result Set
6	The XDS Façade writes an audit trail entry	See Logical Use Case: Write Audit Trail Entry
7	The XDS Façade responds to the requestor with the metadata of the document that match the query and the requestors permissions	See Section B.2


User App (Patient/Doctor)
PHR (via XDS Facade)
Identity Provider
Authorization Service
Audit Trail Service
ref
LUC: Authenticate
Document Query
IHE ITI-18: RegistryStoredQuery-b
ref
LUC: Validate Requestor Authenticity and Authorization
obtain metadata for all documents that match the query
ref
LUC: Filter Result Set
ref
LUC: Write Audit Trail Entry
document metadata
render metadata for display
User App (Patient/Doctor)
PHR (via XDS Facade)
Identity Provider
Authorization Service
Audit Trail Service

B.1.4.1 Filter Result Set

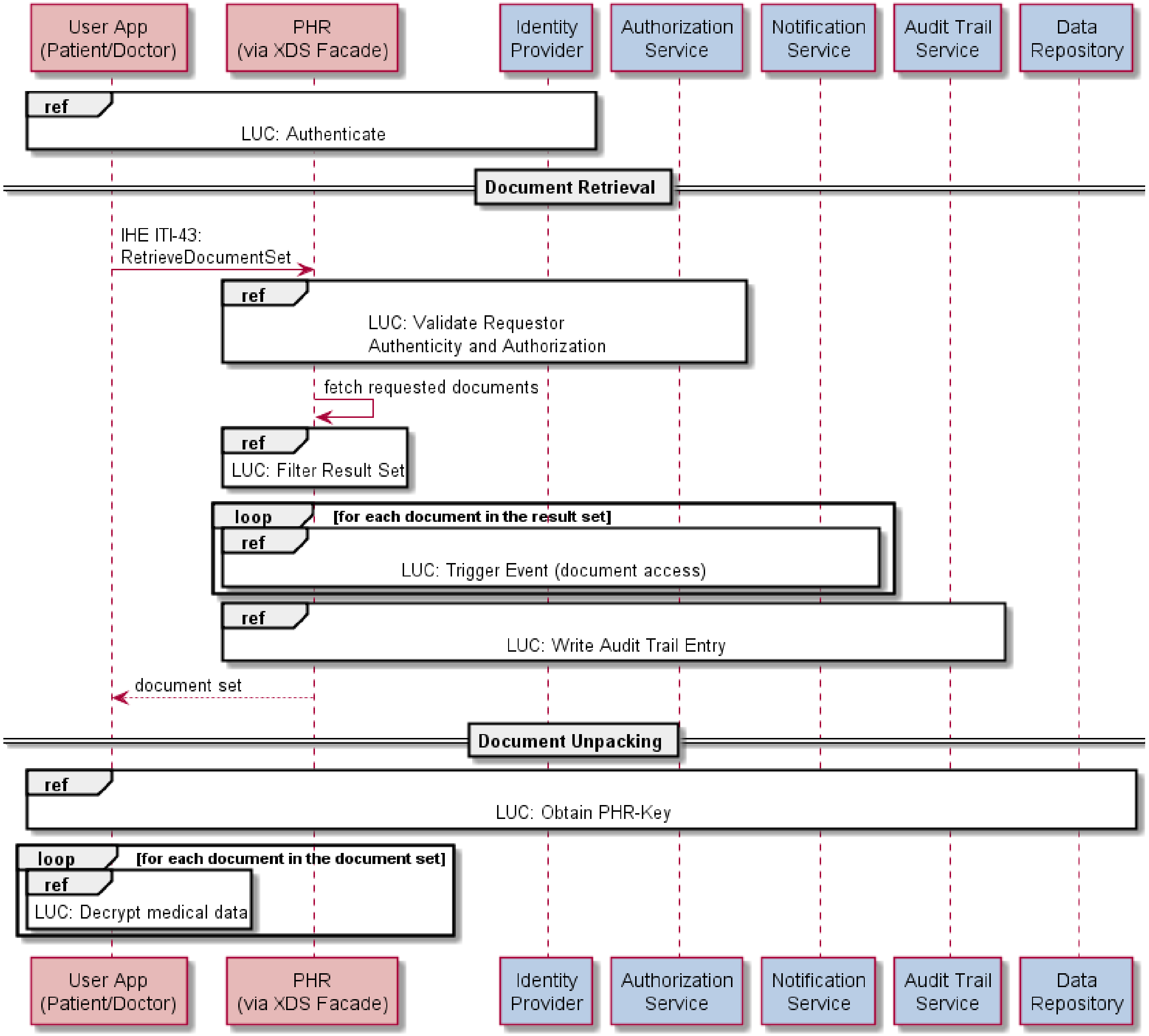
Use Case Name	Filter Result Set
ID	E.HLT-LUC-031
Main Actor	- XDS Façade
Secondary Actors	- Authorization Service
Pre-conditions	- A PHR read access requests has been accepted and processed by the PHR Façade
Post-conditions	- The access requests is validated with respect to the permission of the requestor to receive the documents matching the request
Description	For each document (metadata) in the result set, the XDS Façade validates the authorization of the requestor to receive this document.
Image	 XDS Facade Authorization Service loop [for each document (metadata) in the result set] ref obtain information required for policy decision request by using internal means of the IHE XDS implementation (see section 6.2.2) authorize (D5.1#22) policy decision alt [policy decision is "deny"] remove document (metadata) from result set XDS Facade Authorization Service

B.1.5 Logical Use Case: Retrieve Documents

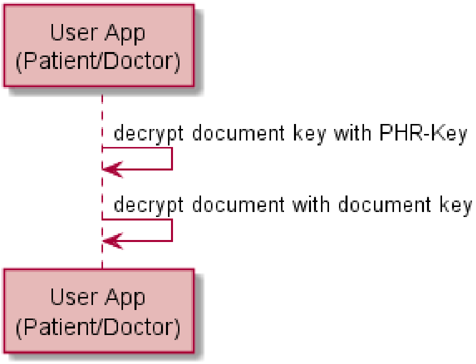
This use case defines functionality for fetching selected documents from the Personal Health Record (PHR). It is assumed that the documents have been identified beforehand through the Logical Use Case: Query Documents. As a result, this use case provides the documents that were selected by the user and comply with the permissions assigned to the user.

#	Flow of Control and Data	Definition
1	The user (patient or doctor) logs in to CREDENTIAL by authenticating with the CREDENTIAL Identity Provider.	See Logical Use Case: Authentication
2	The Patient/Doctor App sends an ITI-43 message to the XDS Façade for retrieving as set of identified documents.	See Section B.2
3	The XDS Façade validates the authenticity and authorization of the requestor.	B.1.3.2: Validate Requestor Authenticity and Authorization

4	The XDS Façade queries the Document Repository for all requested documents	See Section 6.2.2
5	The XDS Façade filters out all documents that the requestor is not authorized to access	6.2.1.4.1: Filter Result Set
6	The XDS Façade triggers the Notification Service to notify the user about a document having been accessed	See Logical Use Case: Trigger Event
7	The XDS Façade writes an audit trail entry	See Logical Use Case: Write Audit Trail Entry
8	The XDS Façade responds to the requestor with the documents that match the query and the requestors permissions	See Section B.2
9	If not the patient is the requestor: The requestor obtains the PHR-Key for the patient's PHR	See Logical Use Case: Obtain PHR-Key
10	The requestor decrypts the documents using the PHR-Key	B.1.5.1: Decrypt medical data


User App (Patient/Doctor)
PHR (via XDS Facade)
Identity Provider
Authorization Service
Notification Service
Audit Trail Service
Data Repository
ref
LUC: Authenticate
Document Retrieval
IHE ITI-43: RetrieveDocumentSet
ref
LUC: Validate Requestor Authenticity and Authorization
fetch requested documents
ref
LUC: Filter Result Set
loop [for each document in the result set]
ref
LUC: Trigger Event (document access)
ref
LUC: Write Audit Trail Entry
document set
Document Unpacking
ref
LUC: Obtain PHR-Key
loop [for each document in the document set]
ref
LUC: Decrypt medical data

B.1.5.1 Decrypt medical data

Use Case Name	Decrypt medical data
ID	E.HLT-LUC-041
Main Actor	- CREDENTIAL App (Patient App or Doctor App)
Secondary Actors	- User (patient or doctor)
Pre-conditions	- User has the CREDENTIAL eHealth App installed and activated - Encrypted document has been retrieved for the patient's PHR - PHR-Key for the source PHR is accessible to the User App
Post-conditions	- Document is ready for display and/or further processing
Description	The user's app decrypts a medical document that has been retrieved from the patient's PHR.
Image	 User App (Patient/Doctor) decrypt document key with PHR-Key decrypt document with document key User App (Patient/Doctor)

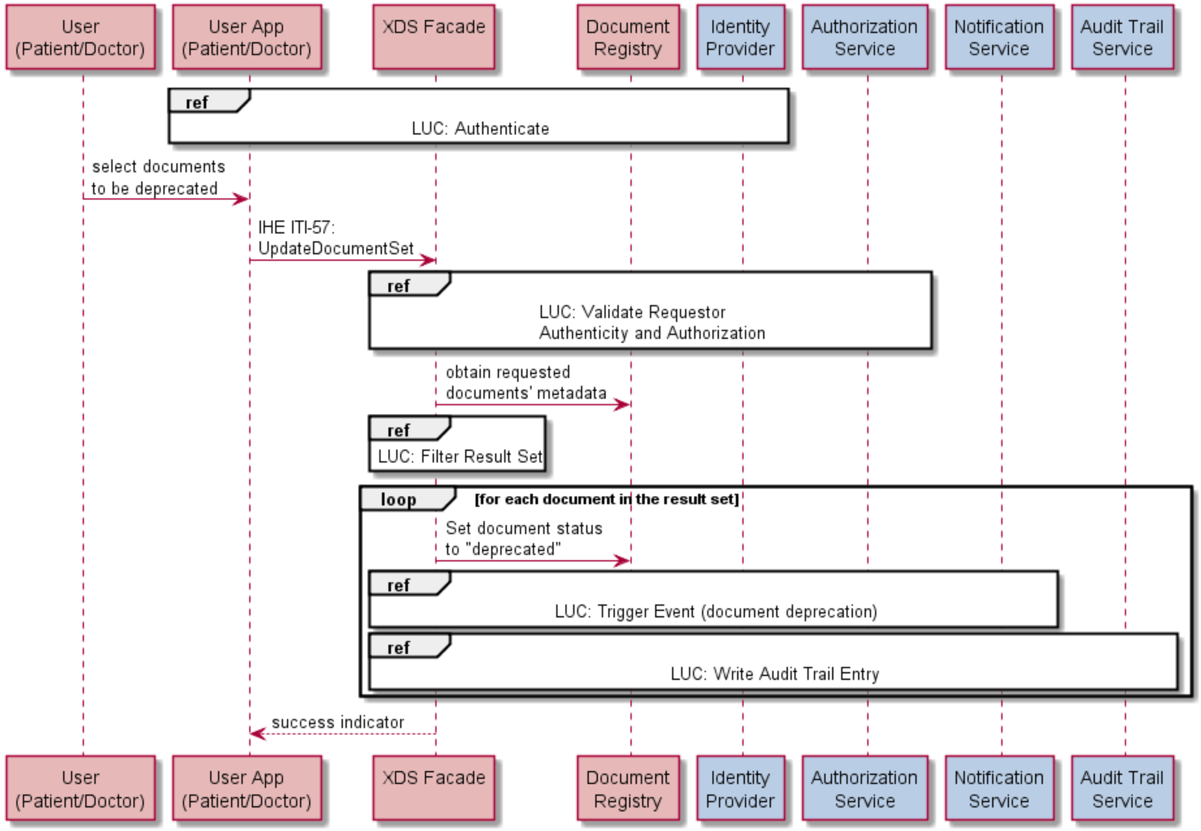
B.1.6 Logical Use Case: Deprecate Documents

This use case provides means to deprecate identified documents within a patient's PHR. It is used in case a document gets outdated or is considered as faulty. Documents of status "deprecated" will no longer be considered by document queries (see Logical Use Case: Query Documents). It is assumed that the documents to be deprecated have been identified beforehand through the Logical Use Case: Query Documents.

The generic flow of control is as follows:

#	Flow of Control and Data	Definition
1	The user (patient or doctor) logs in to CREDENTIAL by authenticating with the CREDENTIAL Identity Provider.	See Logical Use Case: Authentication
2	The user selects the document(s) to be deprecated.	Implementation specific flow depending on the user interaction patterns
3	The Patient/Doctor App sends an ITI-57 message to the XDS Façade.	See Section B.2
4	The XDS Façade validates the authenticity and authorization of the requestor.	B.1.3.2: Validate Requestor Authenticity and Authorization
5	The XDS Façade filters out all document entries that the requestor is not authorized to access.	B.1.4.1: Filter Result Set
6	The XDS Façade deprecates the identified document entries	See Section B.2

7	The XDS Façade triggers the Notification Service to notify the user about a document having been deprecated	See Logical Use Case: Trigger Event
8	The XDS Façade writes an audit trail entry	See Logical Use Case: Write Audit Trail Entry
9	The XDS Façade sends a confirmation back to the requestor. The GUI provides respective feedback to the user.	Implementation specific flow


User (Patient/Doctor)
User App (Patient/Doctor)
XDS Facade
Document Registry
Identity Provider
Authorization Service
Notification Service
Audit Trail Service
ref
LUC: Authenticate
select documents to be deprecated
IHE ITI-57: UpdateDocumentSet
ref
LUC: Validate Requestor Authenticity and Authorization
obtain requested documents' metadata
ref
LUC: Filter Result Set
loop
[for each document in the result set]
Set document status to "deprecated"
ref
LUC: Trigger Event (document deprecation)
ref
LUC: Write Audit Trail Entry
success indicator

B.1.7 Logical Use Case: Authenticate

This use case provides means to authenticate a user and issues a CREDENTIAL identity assertion that can be consumed by CREDENTIAL business and security services.

From a logical perspective, the authentication of the patient or a doctor in the CREDENTIAL eHealth pilot does not impose specific requirements or constraints on the generic CREDENTIAL authentication service. Therefore, the definitions given in CREDENTIAL Deliverable D2.1 “Scenarios and Use Cases” apply to the eHealth pilot, too.

B.1.8 Logical Use Case: Register PHR-Key

This use case registers a PHR-Key by storing it as a document to the CREDENTIAL Wallet. It imposes a specific semantics on top of the generic CREDENTIAL service for storing documents to a CREDENTIAL account:

- The provided document only contains a PHR-Key
- This is the only document stored to a CREDENTIAL eHealth account (except audit trail entries and re-encrypted instances of the document reflecting authorizations given to doctors)

Beside this this use case is identical with the use case "store document to wallet" as defined in CREDENTIAL Deliverable D2.1 "Scenarios and Use Cases"

B.1.9 Logical Use Case: Obtain PHR Key

This use case makes the PHR-Key available to an authorized user. The PHR-Key is needed for decrypting received medical documents and for encrypting medical documents before providing them to a patient's PHR.

This use case is fully identical with the generic CREDENTIAL use case for retrieving an identified document from the CREDENTIAL Wallet (see LUC "retrieve" on Participant Data Directory as defined in D5.1 "Functional Design").

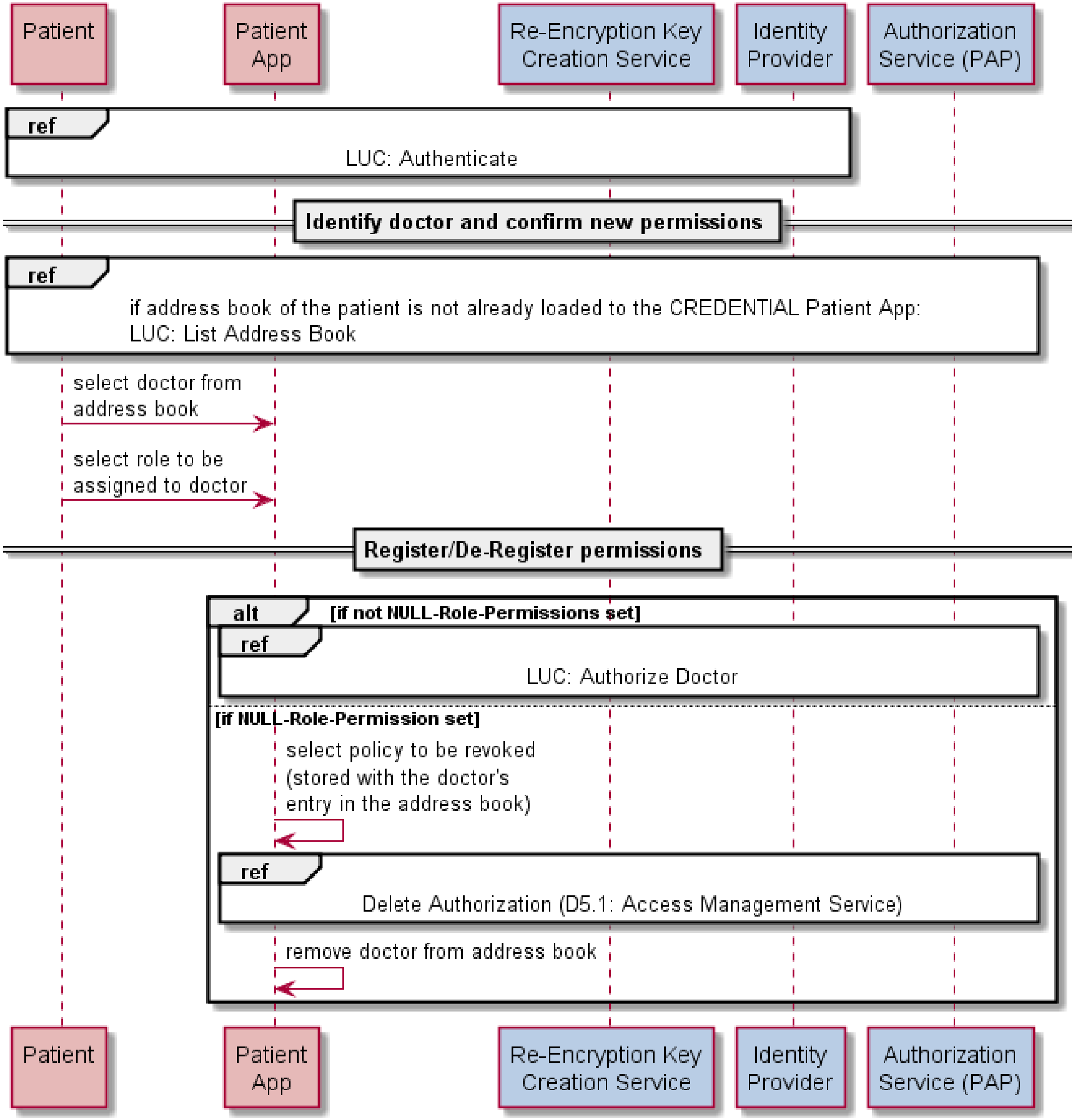
B.1.10 Logical Use Case: Set User Role

This use case can only be triggered by the patient. It assigns a role (e.g., Family Doctor) to an identified user and grants all permissions linked with this role. It can as well be used for taking permissions from a user by assigning this user a role with less permission or by assigning the user a NULL-role (see Section 5.2.6.2 for details).

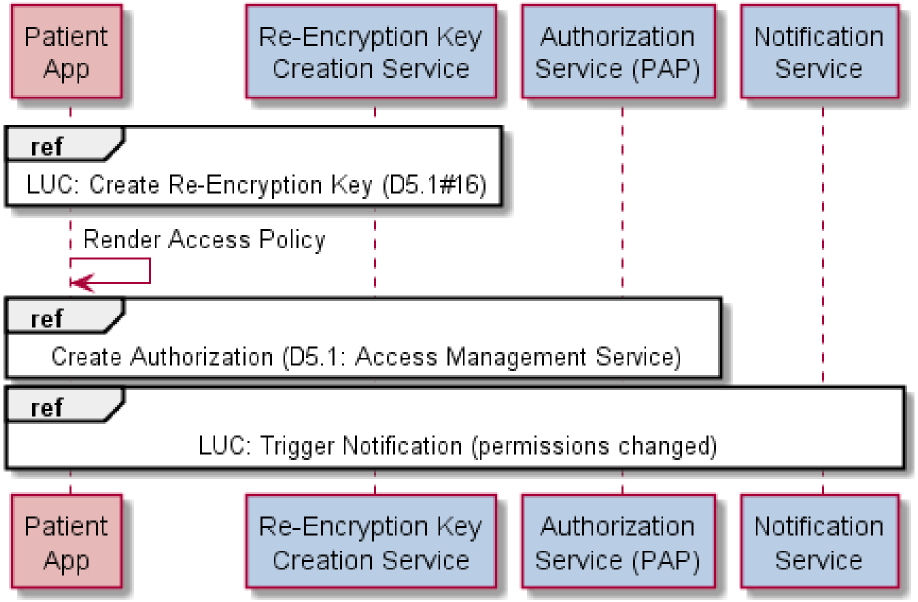
The generic flow of control is as follows:

#	Flow of Control and Data		Definition
1	The patient logs in to CREDENTIAL by authenticating with the CREDENTIAL Identity Provider.		See Logical Use Case: Authentication
2	The patient selects a user from the address book managed by the CREDENTIAL Patient App. If the address book is not loaded, it is generated from the policies registered by the patient.		See Logical Use Case: List Doctors
3	The user selects the role to be assigned to the user.		Implementation specific flow depending on the user interaction patterns
If not a NULL-Role is set:			
	4a	The CREDENTIAL Patient App assembles a policy that reflects the granted permissions on the PHR and gives the selected doctor read-access to the PHR-Key in the CREDENTIAL Wallet.	Implementation specific, e.g., by filling in an XACML template
	5a	The Patient App issues a re-encryption key for the doctor and triggers the authorization of the doctor for accessing his PHR.	See B.1.10.1: Authorize Doctor
If a NULL-Role is set:			
	4b	The CREDENTIAL Patient App requests a list of all policies registered by the user	See D5.1: Policy Administration Point "list" Service

	5b	The CREDENTIAL Patient App selects the policy that granted access permissions to the PHR to the selected user	
	6b	The CREDENTIAL Patient App requests the Policy Administration Point to remove all permissions from the selected user	See D5.1: Policy Administration Point "delete" Service
	7b	The CREDENTIAL Patient App removes the doctor from the patient's address book	


Patient
Patient App
Re-Encryption Key Creation Service
Identity Provider
Authorization Service (PAP)
ref
LUC: Authenticate
Identify doctor and confirm new permissions
ref
if address book of the patient is not already loaded to the CREDENTIAL Patient App:
LUC: List Address Book
select doctor from address book
select role to be assigned to doctor
Register/De-Register permissions
alt
[if not NULL-Role-Permissions set]
ref
LUC: Authorize Doctor
[if NULL-Role-Permission set]
select policy to be revoked (stored with the doctor's entry in the address book)
ref
Delete Authorization (D5.1: Access Management Service)
remove doctor from address book
Patient
Patient App
Re-Encryption Key Creation Service
Identity Provider
Authorization Service (PAP)

B.1.10.1 Authorize Doctor

Use Case Name	Authorize Doctor
ID	E.HLT-LUC-091
Main Actor	- CREDENTIAL Patient App
Secondary Actors	- Authorization Service - Re-Encryption Key Generation Service
Pre-conditions	- Patient has the CREDENTIAL Patient App installed and activated - Patient is authenticated with the App and with CREDENTIAL - Doctor whose permissions are to be changes is identified - Patient has confirmed the permission changes to be applied
Post-conditions	- New permissions are registered for a doctor. - Re-Encryption key is available for the doctor for accessing the patient's PHR-Key - The doctor is notified about his permissions
Description	The CREDENTIAL Patient App requests a re-encryption key for the user. It assembles a policy that reflects the granted permissions on the PHR and gives the selected doctor read-access to the PHR-Key in the CREDENTIAL Wallet. By providing the policy and the re-encryption key to the Policy Administration Point the new permissions are activated.
Image	 Patient App Re-Encryption Key Creation Service Authorization Service (PAP) Notification Service ref LUC: Create Re-Encryption Key (D5.1#16) Render Access Policy ref Create Authorization (D5.1: Access Management Service) ref LUC: Trigger Notification (permissions changed) Patient App Re-Encryption Key Creation Service Authorization Service (PAP) Notification Service

B.1.11 Logical Use Case: Authorize (Ad Hoc)

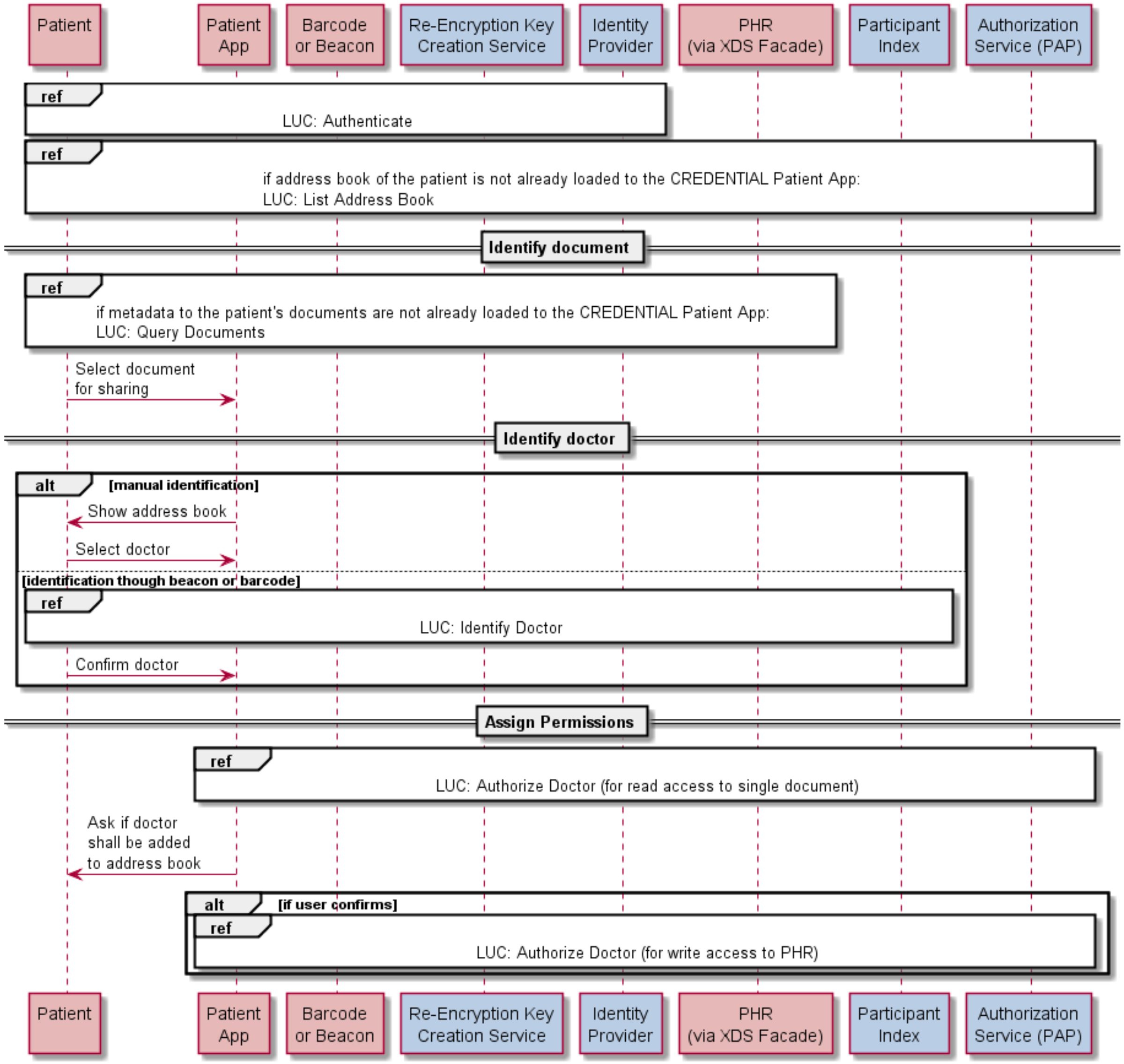
This use case assigns temporary read permissions for a single document to an identified user (see Section 5.2.6.2 for details). Permissions of this kind are only valid for a short period of time and cannot be revoked. For this use case, three scenarios must be considered for identifying the doctor who is to be granted access to a document:

1. The patient selects a document through his Patient App and clicks on the "Share" button. He then selects a doctor from his address book who is granted access to this particular document only.

2. The patient visits a doctor. The doctor likes to get access to a particular document that is available in the patient’s PHR. The patient selects the document through his Patient App and then scans a 2D-barcode that is placed in the doctor’s office. The Patient App identifies the doctor and grants him access to this particular document.
3. The patient visits a doctor. The doctor likes to get access to a particular document that is available in the patient’s PHR. The patient selects the document through his Patient App and then accepts a signal from a beacon that is placed in the doctor’s office. The Patient App identifies the doctor and grants him access to this particular document.

The generic flow of control is as follows:

#	Flow of Control and Data		Definition
1	The patient logs in to CREDENTIAL by authenticating with the CREDENTIAL Identity Provider.		See Logical Use Case: Authentication
2	The patient makes the metadata of the document to be shared available to the CREDENTIAL Patient App (depending on the implementation, the App may even automatically read the table of content of the patient’s PHR after the successful authentication of the patient).		See Logical Use Case: Query Documents
3	The patient selects the document to be shared.		
	3a	Option 1: The patient selects the doctor who is to be given permissions for reading the document from his address book.	Implementation specific, depending on the visualization of the address book and the general interaction patterns
	3b	Option 2: The patient scans a barcode with his smartphone. The CREDENTIAL Patient App uses the information coded in the barcode to identify the doctor who shall be granted access to the selected document.	0: Identify Doctor
	3c	Option 3: The patient accepts a signal that is sent from a beacon to the CREDENTIAL Patient App. The CREDENTIAL Patient App uses the information coded in the beacon signal to identify the doctor who shall be granted access to the selected document.	0: Identify Doctor
4	The Patient app assembles a policy that grants the selected doctor read access to the selected document and read access to the PHR-Key.		Implementation specific, e.g., by filling in an XACML template
5	The Patient App issues a re-encryption key for the doctor and triggers the authorization of the doctor for accessing the selected document.		See B.1.10.1: Authorize Doctor
6	If the doctor is not registered with the patient’s address book, the patient is asked if he wants to add the doctor to his address book. If the patient confirms, the doctor is granted additional write permissions to the PHR.		See B.1.10.1: Authorize Doctor


Patient
Patient App
Barcode or Beacon
Re-Encryption Key Creation Service
Identity Provider
PHR (via XDS Facade)
Participant Index
Authorization Service (PAP)
ref
LUC: Authenticate
ref
if address book of the patient is not already loaded to the CREDENTIAL Patient App:
LUC: List Address Book
Identify document
ref
if metadata to the patient's documents are not already loaded to the CREDENTIAL Patient App:
LUC: Query Documents
Select document for sharing
Identify doctor
alt
[manual identification]
Show address book
Select doctor
[identification though beacon or barcode]
ref
LUC: Identify Doctor
Confirm doctor
Assign Permissions
ref
LUC: Authorize Doctor (for read access to single document)
Ask if doctor shall be added to address book
alt
[if user confirms]
ref
LUC: Authorize Doctor (for write access to PHR)
Patient
Patient App
Barcode or Beacon
Re-Encryption Key Creation Service
Identity Provider
PHR (via XDS Facade)
Participant Index
Authorization Service (PAP)

B.1.12 Logical Use Case: Validate Permissions

This logical use case performs the assessment of an access request with respect to the policies as defined by the patient. The use case fully corresponds to the generic logical use case “authenticate” as defined as a service to the CREDENTIAL Authorization Service (see D5.1 “Functional Design”).

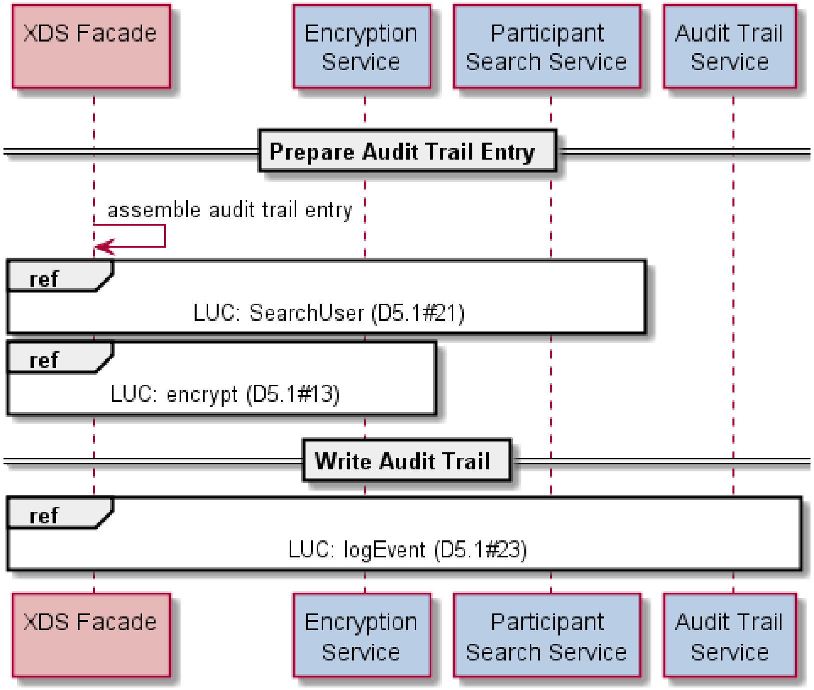
B.1.13 Logical Use Case: Write Audit Trail Entry

While generic CREDENTIAL logical use case utilized by the eHealth pilot inherently log all auditable events, PHR services explicitly need to execute this use case for writing an audit trail entry to the CREDENTIAL Audit Trail Service.

As audit trail entries written by the PHR services contain protected personal information (e.g., which doctors accessed data for a patient), all audit trail entries written by the PHR services are encrypted through means of CREDENTIAL. An encryption scheme is used, that only allows the patient to access his audit trail or to grant permissions to privacy commissioners using CREDENTIAL proxy re-encryption.

The generic flow of control is as follows:

#	Flow of Control and Data	Definition
1	The PHR Facade assembles an audit trail entry for an event according to the definitions of the underlying IHE transaction.	See B.2 for considerations on audit trail entries for the IHE transactions used by the eHealth pilot
2	The PHR Façade obtains the patient's public key from the CREDENTIAL Participant Search Service.	See LUC SearchUser (D5.1, Participant Search Service)
3	The PHR Façade encrypts the audit trail using the patient's public key.	See LUC Encrypt (D5.1, Encryption Service)
4	The PHR Façade writes the encrypted audit trail entry to the CREDENTIAL Audit Trail Service. The entry itself is wrapped by a CREDENTIAL-compliant envelope that holds the account id and a flag that marks this entry as PHR relevant.	See LUC LogEvent (D5.1, Audit Trail Service).

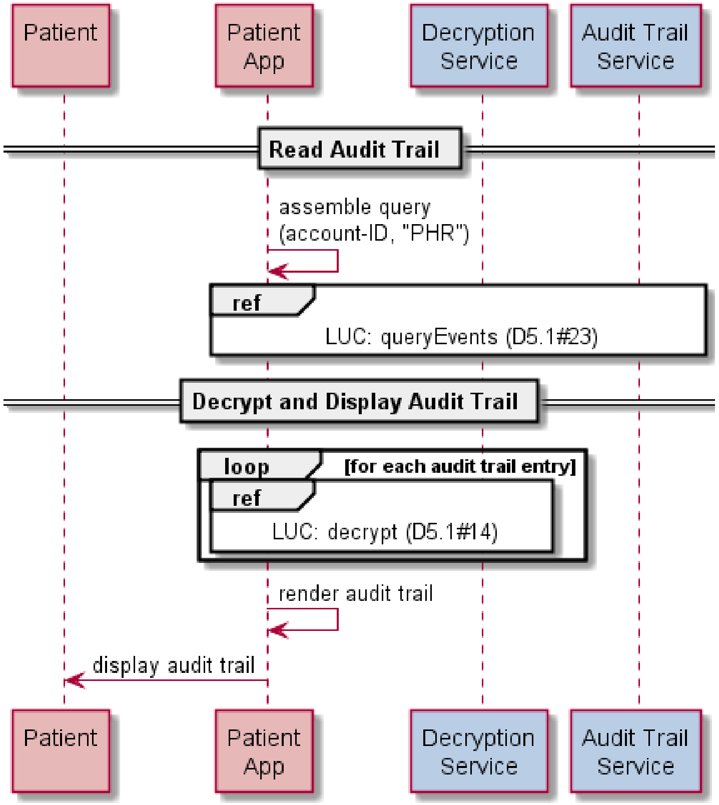


B.1.14 Logical Use Case: Read Audit Trail

This use case defines how to read audit trail entries relevant for an identified PHR from the CREDENTIAL Audit Trail Service. While this use case uses the generic CREDENTIAL functionality for discovering and reading relevant audit trail entries, it adds functionality for decrypting the formerly encrypted entries. Decryption can only be performed by the patient. Nevertheless the patient may use CREDENTIAL proxy re-encryption services for making an audit trail entry readable to other persons.

The generic flow of control is as follows:

#	Flow of Control and Data	Definition
1	The CREDENTIAL Patient App reads the audit trail entries associated to a defined CREDENTIAL account from the Audit Trail Service. The given query is only on entries linked to the patient's account and marked as PHR relevant (see Logical Use Case "Write Audit Trail Entry".)	See LUC queryEvents (D5.1, Audit Trail Service).
3	The CREDENTIAL Patient App decrypts the audit trail entries using the local CREDENTIAL service that manages the patient's private key.	See LUC Decrypt (D5.1, Decryption Service)
4	The CREDENTIAL Patient App renders the audit trail for display and/or further processing.	Implementation specific, depending on the functionality implemented on top of an audit trail.



B.1.15 Logical Use Case: Decrypt PHR-Key

This use case fully complies with the generic use case for decrypting a document that had been encrypted by means of CREDENTIAL. The document provided for decryption contains the PHR-Key for accessing medical data linked to an identified account.

For details on this use case see the definition of the generic Decryption Service as specified in D5.1 “Functional Design”.

B.1.16 Logical Use Case: Re-Encrypt PHR-Key

This use case fully complies with the generic use case for proxy re-encryption of a document that had been encrypted by means of CREDENTIAL. The document provided for re-encryption contains the PHR-Key for accessing medical data linked to an identified account.

For details on this use case see the definition of the generic Re-Encryption Service as specified in D5.1 “Functional Design”.

B.1.17 Logical Use Case: Add Doctor to Address Book

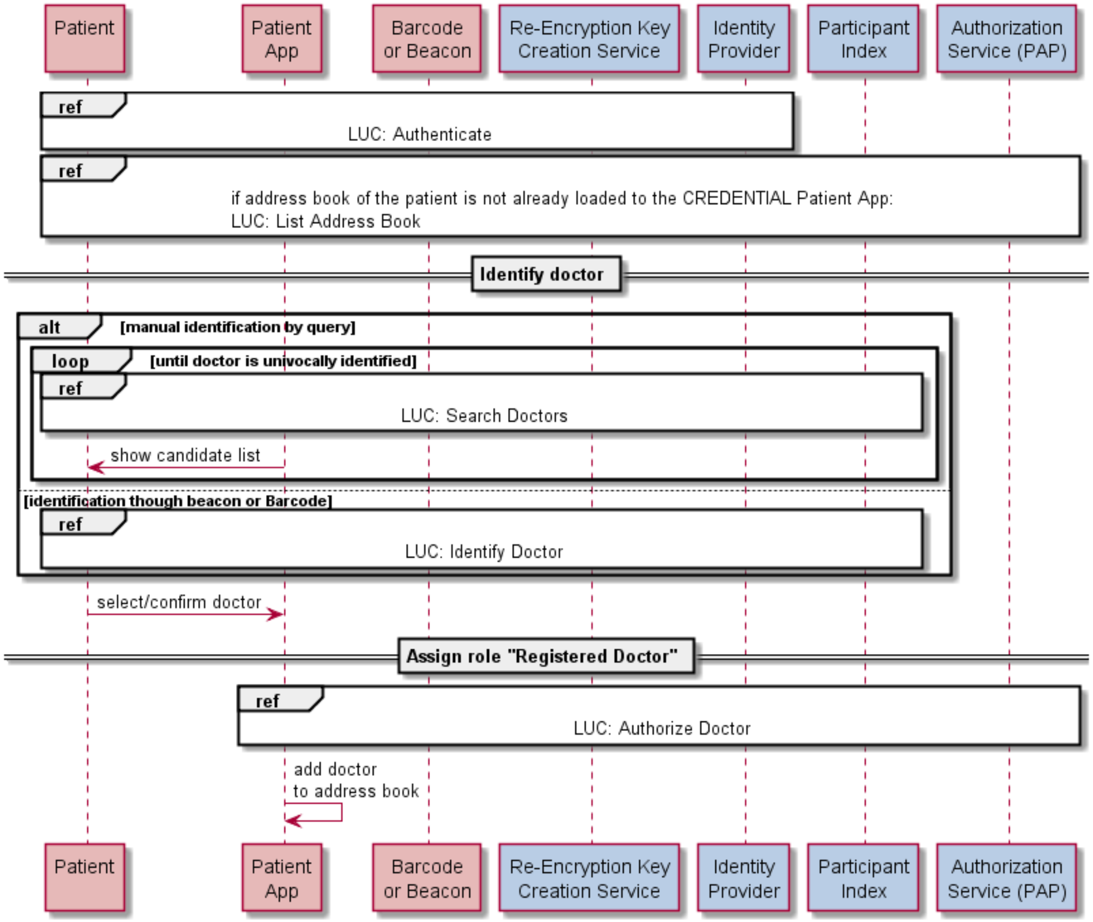
This use case defines how to access the CREDENTIAL participant directory for finding details on a doctor who is not listed in the patient’s address book. . For this use case, three scenarios must be considered for identifying the doctor who is to be added to the address book:

1. The patient connects to the CREDENTIAL Participant Index and searches for the doctor by providing search criteria such as the doctor’s name or profession.
2. The patient visits a doctor. With his smartphone the patient scans a 2D-barcode that is placed in the doctor’s office. The Patient App identifies the doctor through an identifier that is coded within the barcode.
3. The patient visits a doctor. The patient’s CREDENTIAL Patient App accepts a signal from a beacon that is placed in the doctor’s office. The Patient App identifies the doctor through an identifier that is coded within the signal.

The generic flow of control (considering these options) is as follows:

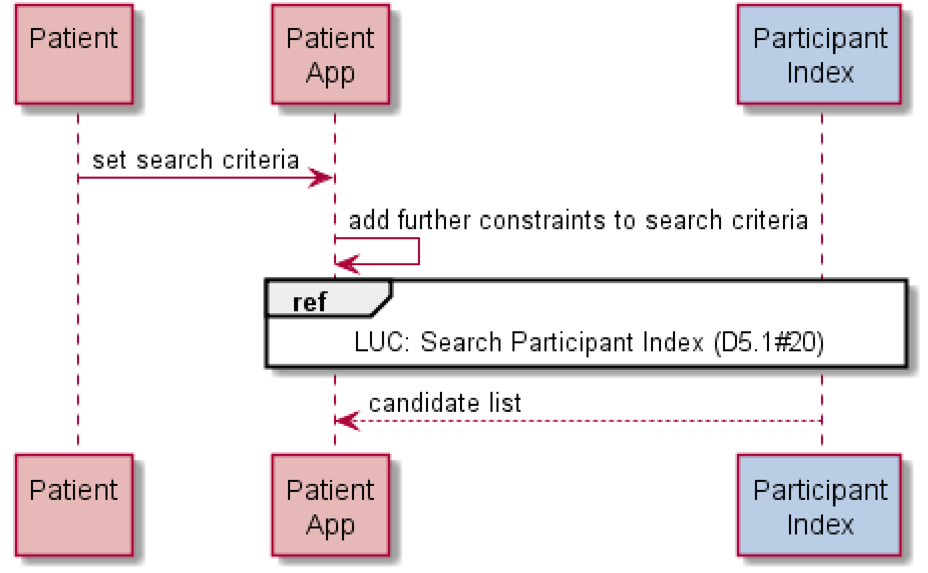
#	Flow of Control and Data		Definition
1	The patient logs in to CREDENTIAL by authenticating with the CREDENTIAL Identity Provider.		See Logical Use Case: Authentication
2	If the address book is not already loaded, it is generated from the policies registered by the patient.		See Logical Use Case: List Doctors
	Option 1: Manual selection of a doctor		
	3a	The patient provides sufficient information for searching the doctor he wants to add to his address book (e.g., clinical discipline, name, address).	Implementation specific depending on the user interaction patterns
	4a	The CREDENTIAL Patient App obtains a list of all doctors and organizations matching the query parameters from the CREDENTIAL Participant Index.	B.1.17.1: Search Doctors

	5a	The patient selects a doctor or organization from the list which shall be added to the address book (alternatively the patient steps back to step 3 and modified his search criteria)		Implementation specific depending on the user interaction paradigms as implemented by the App's GUI
	Option 2: Automated selection of a doctor			
		3b1	Option 2a: The patient scans a barcode with his smartphone. The CREDENTIAL Patient App extracts the doctor's CREDENTIAL ID from the barcode.	0: Identify Doctor
		3b2	Option 2b: The patient accepts a signal that is sent from a beacon to the CREDENTIAL Patient App. The CREDENTIAL Patient App extracts the doctor's CREDENTIAL ID from the barcode.	0: Identify Doctor
	5b	The CREDENTIAL Patient App loads detailed information about the doctor from the CREDENTIAL Participant Index. The App asks the patient to confirm that this is the doctor he wanted to select.		0: Identify Doctor
6	The CREDENTIAL Patient App assigns the role "Registered Doctor" to the selected doctor (or organization).			B.1.10.1: Authorize Doctor
7	The CREDENTIAL Patient App adds the doctor to the address book.			Implementation specific depending on the visualization of the address book

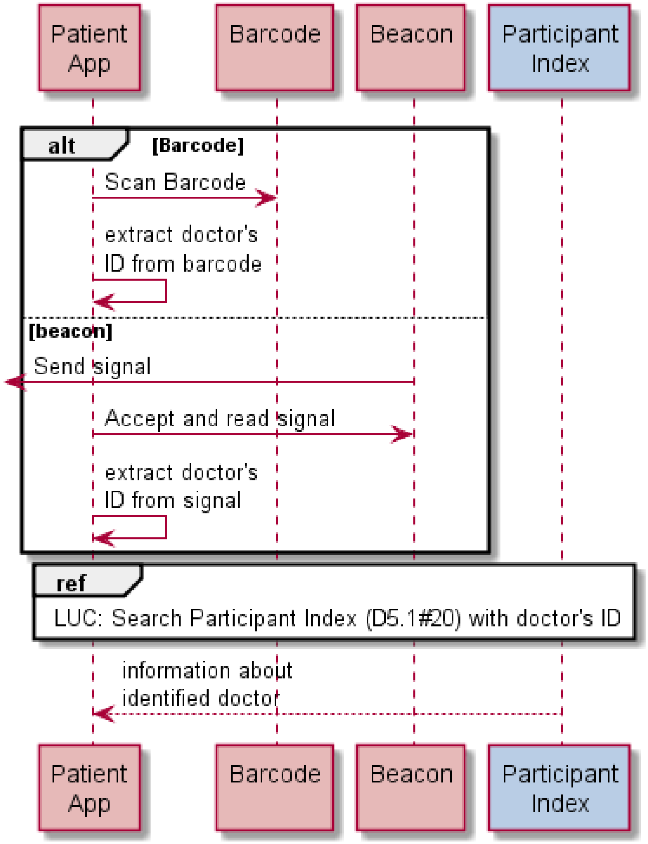

Patient
Patient App
Barcode or Beacon
Re-Encryption Key Creation Service
Identity Provider
Participant Index
Authorization Service (PAP)
ref
LUC: Authenticate
ref
if address book of the patient is not already loaded to the CREDENTIAL Patient App:
LUC: List Address Book
Identify doctor
alt
[manual identification by query]
loop
[until doctor is univocally identified]
ref
LUC: Search Doctors
show candidate list
[identification though beacon or Barcode]
ref
LUC: Identify Doctor
select/confirm doctor
Assign role "Registered Doctor"
ref
LUC: Authorize Doctor
add doctor
to address book
Patient
Patient App
Barcode or Beacon
Re-Encryption Key Creation Service
Identity Provider
Participant Index
Authorization Service (PAP)

B.1.17.1 Search Doctors

Use Case Name	Search Doctors
ID	E.HLT-LUC-151
Main Actor	- CREDENTIAL Patient App
Secondary Actors	- Participant Index
Pre-conditions	- Patient has the CREDENTIAL Patient App installed and activated - Patient is authenticated with the App and with CREDENTIAL
Post-conditions	- A candidate list of doctors who match the patient's query is obtained from the CREDENTIAL Participant Index
Description	The patient looks up the CREDENTIAL Participant Index for a certain doctor or healthcare organization. The following kinds of query will be supported: - Search for details on a doctor whose CREDENTIAL Account ID is known to the user - Search by name and city (e.g., Dr. Kopfer in Berlin) - Search by profession and city (e.g., cardiologist in Berlin) The Patient App takes responsibility that a query only provides entries for doctors and healthcare organizations. In order to ensure this it may add further constraints to the query (e.g., "role = doctor").

Image	Patient Patient App Participant Index set search criteria add further constraints to search criteria ref LUC: Search Participant Index (D5.1#20) candidate list Patient Patient App Participant Index

B.1.17.2 Identify Doctor

Use Case Name	Identify Doctor
ID	E.HLT-LUC-152
Main Actor	- CREDENTIAL Patient App
Secondary Actors	- Barcode or Beacon - Participant Index
Pre-conditions	- Patient has the CREDENTIAL Patient App installed and activated - Patient is authenticated with the App and with CREDENTIAL - In the doctor's office a beacon or barcode is available for obtaining the doctor's CREDENTIAL ID
Post-conditions	- A doctor is univocally identified and detailed information about this doctor is available to the CREDENTIAL Patient App
Description	Use of a barcode: The patient scans a barcode with his smartphone. The CREDENTIAL Patient App extracts the doctor's CREDENTIAL ID from the barcode. Use of a beacon: The patient accepts a signal that is sent from a beacon to the CREDENTIAL Patient App. The CREDENTIAL Patient App extracts the doctor's CREDENTIAL ID from the barcode. The CREDENTIAL Patient App loads detailed information about the doctor from the CREDENTIAL Participant Index. The App asks the patient to confirm that this is the doctor he wanted to select.
Image	

Patient App
Barcode
Beacon
Participant Index
alt
[Barcode]
Scan Barcode
extract doctor's ID from barcode
[beacon]
Send signal
Accept and read signal
extract doctor's ID from signal
ref
LUC: Search Participant Index (D5.1#20) with doctor's ID
information about identified doctor
Patient App
Barcode
Beacon
Participant Index

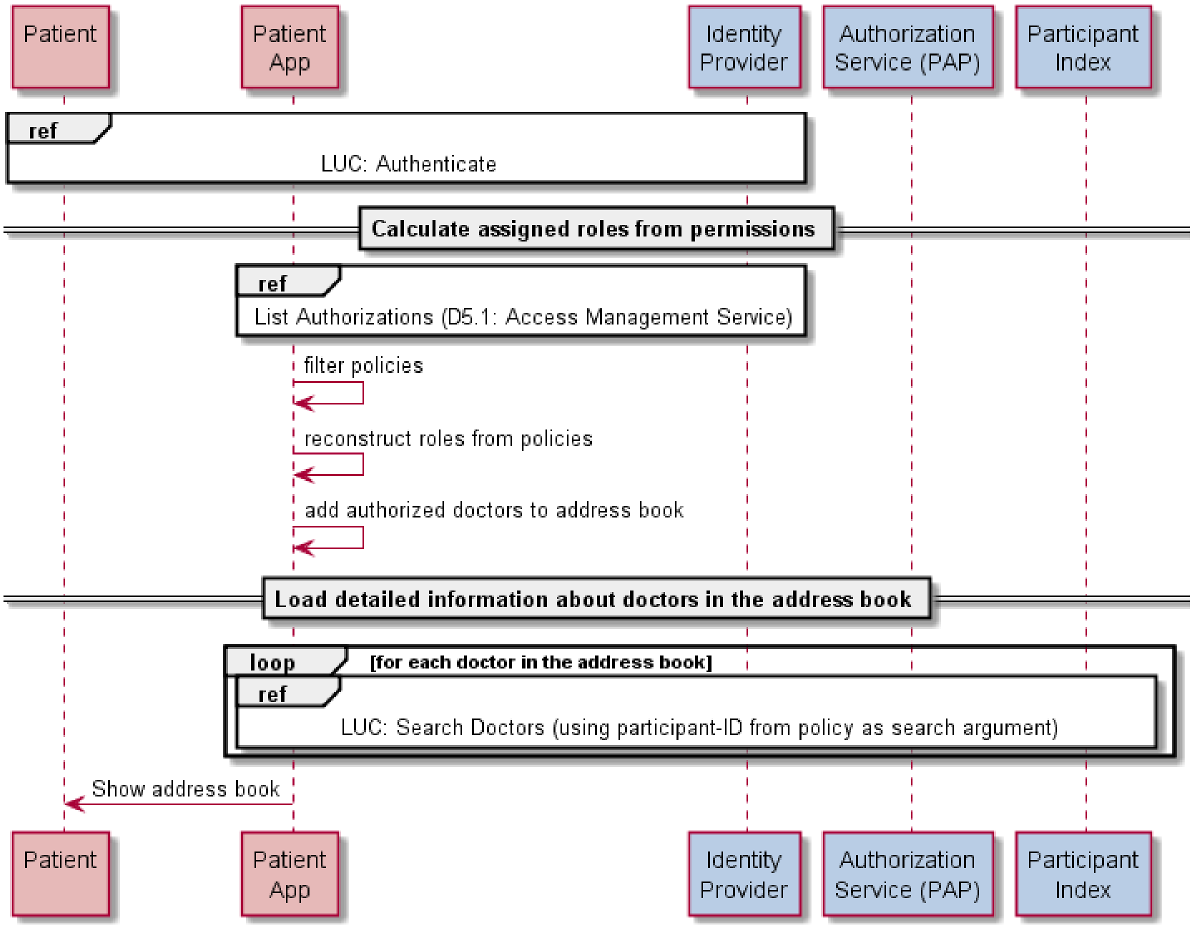
B.1.18 Logical Use Case: List Address Bok

The CREDENTIAL Patient App maintains the patient's address book where the patient's doctors are listed. All doctors listed in the address book at least have full write access to the patient's PHR. In order to not persistently store any protected information with the CREDENTIAL Patent App, the address book is dynamically generated from the permissions given to doctors by the patient.

This use case defines how the CREDENTIAL Patient App dynamically loads the patient's address book using generic CREDENTIAL services.

The generic flow of control is as follows:

#	Flow of Control and Data	Definition
1	The patient logs in to CREDENTIAL by authenticating with the CREDENTIAL Identity Provider.	See Logical Use Case: Authentication
2	The CREDENTIAL Patient App requests the list of all policies set by the patient from the Policy Administration Point.	See D5.1: Policy Administration Point "list" Service
3	The CREDENTIAL Patient App filters out all doctors who are only assigned permissions to access a single document	Details left to the implementation
4	The CEDENTIAL Patient App assigns the role "Registered Doctor" to all doctors in the list.	Details left to the implementation
5	The CREDENTIAL Patient App assigns the additional role "diabetologist" or "family doctor" to all doctors with full access rights (depending on detailed setting from the policy).	Details left to the implementation
6	The CREDENTIAL Patient App obtains details about all listed doctors from the CREDENTIAL Participant Index.	B.1.17.1: Search Doctors
7	The CREDENTIAL Patient App renders the address book for display to the patient.	Implementation specific depending on the visualization of the address book


Patient
Patient App
Identity Provider
Authorization Service (PAP)
Participant Index
ref
LUC: Authenticate
Calculate assigned roles from permissions
ref
List Authorizations (D5.1: Access Management Service)
filter policies
reconstruct roles from policies
add authorized doctors to address book
Load detailed information about doctors in the address book
loop
[for each doctor in the address book]
ref
LUC: Search Doctors (using participant-ID from policy as search argument)
Show address book
Patient
Patient App
Identity Provider
Authorization Service (PAP)
Participant Index

B.1.19 Logical Use Case: Subscribe Event

The CREDENTIAL eHealth Pilot uses the mechanisms of the CREDENTIAL Notification service for subscribing to events which are triggered by the PHR setrvices. The table below lists how PHR events map onto defined CREDENTIAL subscriptions (see D5.1 on Notification Service for details):

PHR Event	CREDENTIAL Subscription
A new document has been added to the PHR	Subscription to the patient's CREDENTIAL account (preference = add)
A document has been updated	Subscription to the patient's CREDENTIAL account (preference = add or preference = update) or subscription to the particular document (preference = update)
A document has been deprecated	Subscription to the patient's CREDENTIAL account (preference = update) or subscription to the particular document (preference = delete)
A document has been downloaded by an authorized user	Subscription to the patient's CREDENTIAL account (preference = read) or subscription to a particular document (preference = read)

Further subscriptions – e.g., on new authorizations – are provided to users by means defined for the generic CREDENTIAL services. By this this eHealth use case fully maps onto already defined CREDENTIAL use cases and interfaces and will not be further elaborated here.

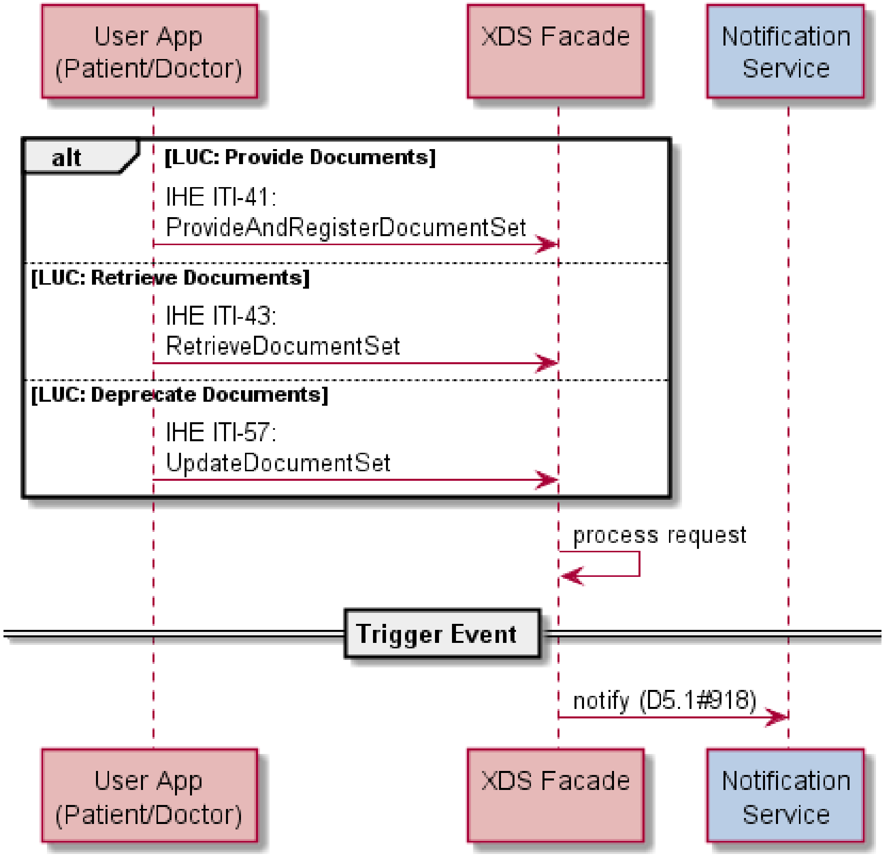
B.1.20 Logical Use Case: Trigger Event

The XDS Facade uses the mechanisms of the CREDENTIAL Notification service for triggering events that originated from PHR services. The table below lists how PHR services trigger CREDENTIAL notifications (see D5.1 on Notification Service for details). Multiple entries for a logical use case (LUC) signal that this LUC triggers more than one CREDENTIAL event.

PHR LUC	CREDENTIAL Event		
	Operation	Target Object Type	Target Object ID
Provide Documents (w/o Update)	add	document	new document ID
Provide Documents (with Update)	add	document	new document ID
	delete	document	old document ID
Retrieve Documents	read	document	document ID
Deprecate Documents	delete	document	document ID

The generic flow of control is as follows:

#	Flow of Control and Data	Definition
1	The XDS Façade receives and processes a request that adds a document to a PHR, updates an existing document, results in a document download or deprecates a document	See LUC Provide Documents (B.1.3), LUC Retrieve Documents (B.1.5), LUC Deprecate Documents (B.1.6)
2	The XDS Façade triggers an event with the CREDENTIAL Notification Service	see “Notify“ Service of the CREDENTIAL Notification Service (D5.1 “Functional Design”)


User App (Patient/Doctor)
XDS Facade
Notification Service
alt
[LUC: Provide Documents]
IHE ITI-41: ProvideAndRegisterDocumentSet
[LUC: Retrieve Documents]
IHE ITI-43: RetrieveDocumentSet
[LUC: Deprecate Documents]
IHE ITI-57: UpdateDocumentSet
process request
Trigger Event
notify (D5.1#918)
User App (Patient/Doctor)
XDS Facade
Notification Service

B.1.21 Logical Use Case: Notify on Event

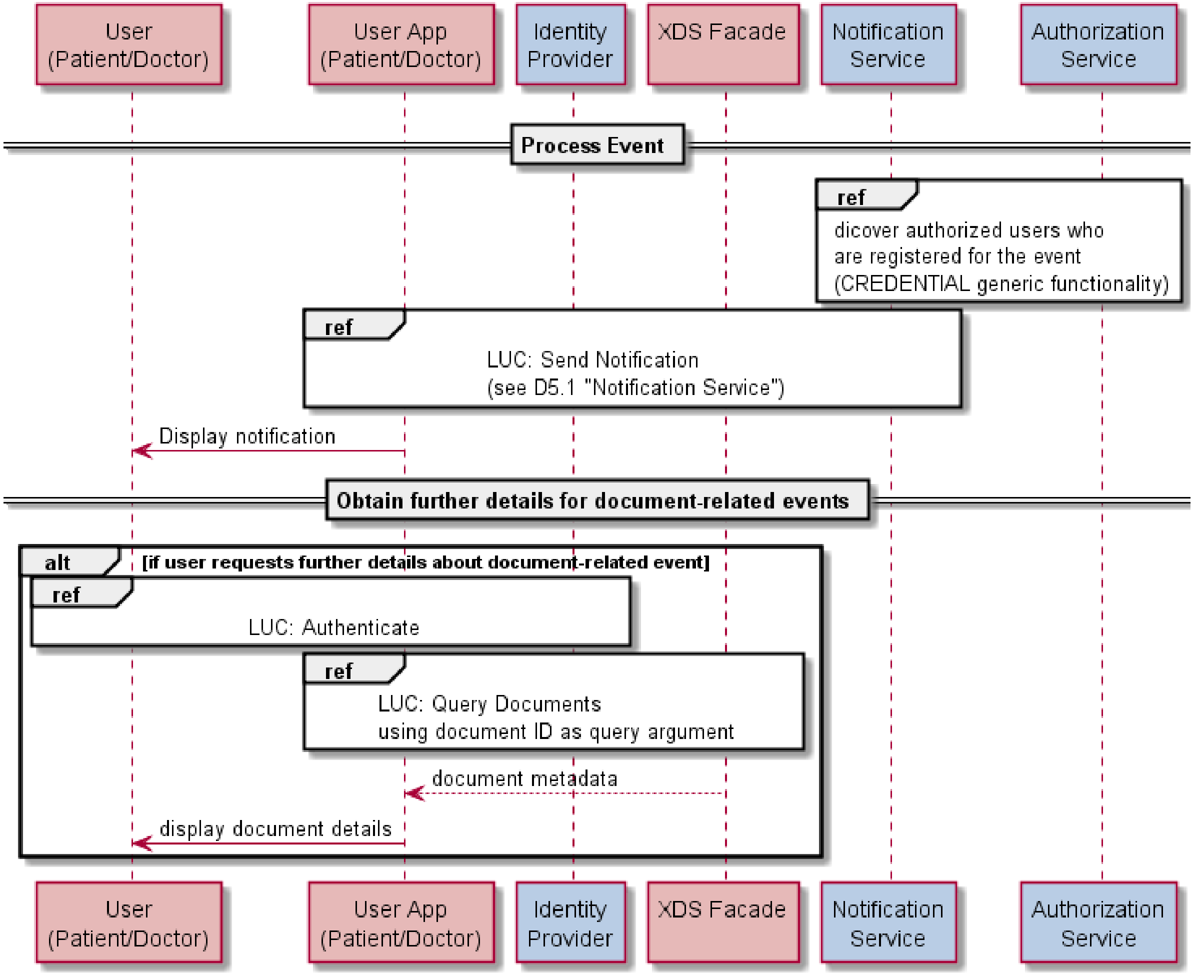
The CREDENTIAL eHealth Pilot uses the mechanisms of the CREDENTIAL Notification service for transmitting event notifications to CREDENTIAL users.

To protect the privacy of the patient, a notification as sent by the CREDENTIAL Notification Service only contains the kind of event (e.g., “new document”) and the identifier of the document that caused the vent. Therefore this use case extends the generic CREDENTIAL logical use case “Send Notification” by securely providing additional information about the affected document upon request by the user.

The generic flow of control is as follows:

#	Flow of Control and Data	Definition
1	The CREDENTIAL Notification Service is informed about a reportable event. It looks up the subscribers to this event and validates the authorization of these users to receive the event.	See D5.1 about the internal functionality of the CREDENTIAL Notification Service
2	The CREDENTIAL Notification Service sends a notification message to the discovered users’ CREDENTIAL App.	LUC Send Notification (see D5.1 “Functional Design”)
3	The CREDENTIAL Patient/Doctor App shows the event notification to the user and asks if the user wants to see more detailed information about the affected document.	Implementation dependent; may vary on different devices and platforms.
If the user confirms to see further details:		

4	The user logs in to CREDENTIAL by authenticating with the CREDENTIAL Identity Provider.	See Logical Use Case: Authentication (B.1.7)
5	The CREDENTIAL Patient/Doctor App sends a query to the PHR for the affected document's metadata.	B.2.8.1: LUC Query Documents
6	The CREDENTIAL Patient/Doctor App renders selected metadata for display to the patient.	Implementation specific depending on the overall GUI design and interaction patterns


User (Patient/Doctor)
User App (Patient/Doctor)
Identity Provider
XDS Facade
Notification Service
Authorization Service
Process Event
ref
dicover authorized users who are registered for the event (CREDENTIAL generic functionality)
ref
LUC: Send Notification (see D5.1 "Notification Service")
Display notification
Obtain further details for document-related events
alt
[if user requests further details about document-related event]
ref
LUC: Authenticate
ref
LUC: Query Documents using document ID as query argument
document metadata
display document details

B.1.22 Logical Use Case: Register Alerts

The core of the diabetes use case chosen for the CREDENTIAL eHealth pilot consists of various monitoring activities, e.g., checking incoming lab results, tracking the patient's compliance and monitoring progress towards the defined care goals. Due to the amount of data such a use case generates for each patient, doctors need automated means for identifying the patients that need immediate response and support.

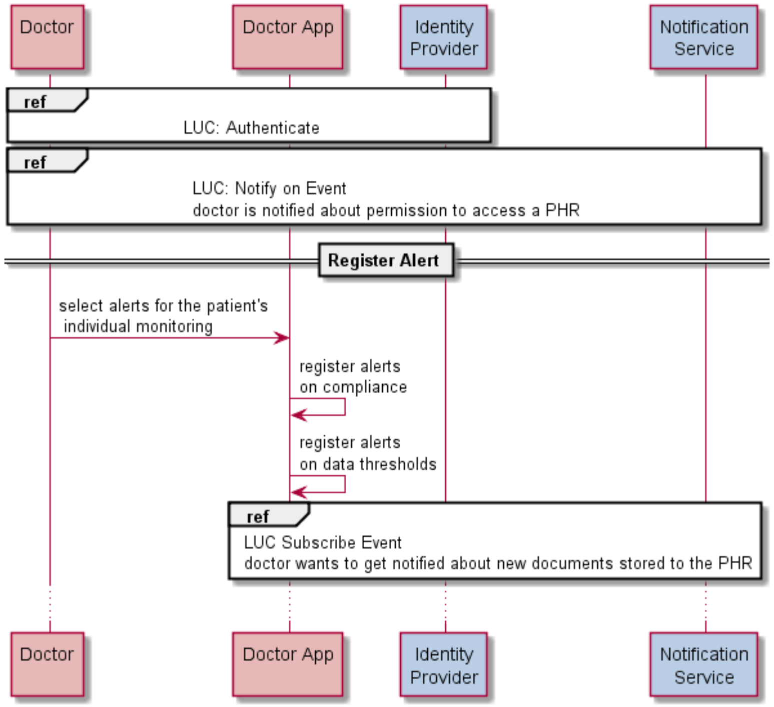
In the eHealth use case this is implemented through alert conditions that can be defined by doctors within their CREDENTIAL Doctor App. Two kinds of alerts are supported:

- Threshold alerts: the doctor may define thresholds for certain data, e.g., for being informed if a patient's HbA1c is above 8.5 or if a patient's weight has increased above 85 kg.
- Compliance alerts: the doctor may define time spans for the availability of data and gets informed if the defined data is not available regularly. For instance, a doctor may set an alert to get informed if a patient did not provide weight values for more than a week.

Alerts are registered individually per patient with the CREDENTIAL Doctor App. For automating the handling of alerts (see next logical use case) an event subscription is issued by the App which allows the system to assess new data without the doctor being forced to manually look for relevamt information.

The generic flow of control for registering alerts is as follows:

#	Flow of Control and Data	Definition
1	The doctor is authenticated with CREDENTIAL and receives a notification that he has been granted access to the PHR of one of his patients.	B.1.21: LUC Notify on Event
2	The CREDENTIAL Doctor App offers the doctor a list of diagnoses related alerts to choose from. The doctor selects the alerts he wants to register and defines the patient's individual thresholds and compliance parameters.	Implementation specific depending on the GUI design and general interaction patterns
3	The Doctor App registers an event with the CREDENTIAL notifications service to receive a notification whenever a document is added to the patient's PHR.	B.1.19: LUC Subscribe Event


Doctor
Doctor App
Identity Provider
Notification Service
ref
LUC: Authenticate
ref
LUC: Notify on Event
doctor is notified about permission to access a PHR
Register Alert
select alerts for the patient's individual monitoring
register alerts on compliance
register alerts on data thresholds
ref
LUC Subscribe Event
doctor wants to get notified about new documents stored to the PHR
Doctor
Doctor App
Identity Provider
Notification Service

B.1.23 Logical Use Case: Handle Alerts

Threshold alerts and compliance alerts require a different handling as

- threshold alerts are triggered by inspecting data which is newly available, while
- compliance alerts are triggered on the non-availability of certain data.

For threshold alerts the generic flow of control for handling alerts is as follows:

#	Flow of Control and Data	Definition
1	The CREDENTIAL Doctor App receives a notification that new data is available to a patient's PHR.	B.1.21: LUC Notify on Event
2	The CREDENTIAL Doctor App logs in to CREDENTIAL on behalf of the doctor.	B.1.7: LUC Authenticate
3	The CREDENTIAL Doctor App fetches the newly provided document from the patient's PHR.	B.1.5: LUC Retrieve Document
4	The App assesses the data in the document against defined thresholds.	Implementation specific depending on the implementation of the alert management
5	If the value of the monitored data element is outside the defined range, a respective message is shown to the doctor.	Implementation specific depending on the means of the platform for displaying messages to users

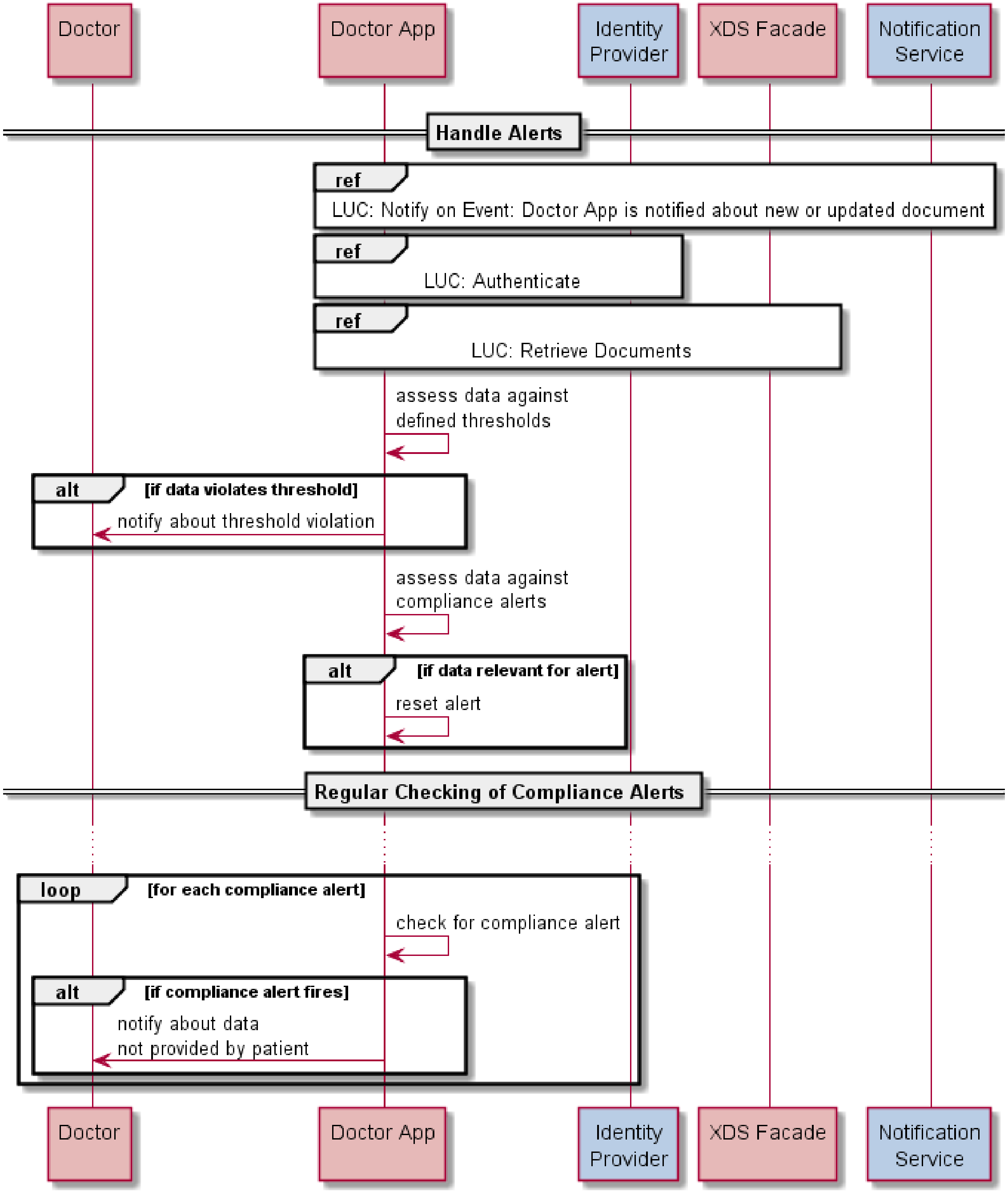
For compliance alerts the generic flow of control for handling alerts is as follows:

#	Flow of Control and Data	Definition
1	The CREDENTIAL Doctor App receives a notification that new data is available to a patient's PHR.	B.1.21: LUC Notify on Event
2	The CREDENTIAL Doctor App logs in to CREDENTIAL on behalf of the doctor.	B.1.7: LUC Authenticate
3	The CREDENTIAL Doctor App fetches the newly provided document from the patient's PHR.	B.2.13.1: LUC Retrieve Document
4	The App assesses the data in the document against defined compliance criteria. If monitored data is available, the respective compliance alert is reset.	Implementation specific depending on the implementation of the alert management

In addition, the CREDENTIAL Doctor App regularly scans through all defined compliance alerts:

#	Flow of Control and Data	Definition
1	The App checks if the time span - within that certain data must have been provided - is reached.	Implementation specific depending on the implementation of the alert management

2	If so, a respective message is shown to the doctor.	Implementation specific depending on the means of the platform for displaying messages to users


Doctor
Doctor App
Identity Provider
XDS Facade
Notification Service
Handle Alerts
ref
LUC: Notify on Event: Doctor App is notified about new or updated document
ref
LUC: Authenticate
ref
LUC: Retrieve Documents
assess data against defined thresholds
alt
[if data violates threshold]
notify about threshold violation
assess data against compliance alerts
alt
[if data relevant for alert]
reset alert
Regular Checking of Compliance Alerts
loop
[for each compliance alert]
check for compliance alert
alt
[if compliance alert fires]
notify about data not provided by patient
Doctor
Doctor App
Identity Provider
XDS Facade
Notification Service

B.2 CREDENTIAL PHR Technical Specification

This section contains the technical specification of the Personal Health Record (PHRI as used for the CREDENTIAL eHealth Pilot. Core of this section is the binding of the CREDENTIAL PHR logical use cases onto IHE XDS actors and transactions. An overview on this mapping is provided in Section B.2.1 while Sections B.2.2 and B.2.3 go into further details of how IHE XDS metadata and messages shall be profiled for CREDENTIAL.

All CREDENTIAL profiles on IHE transactions and metadata definitions take into consideration the IHE European Interoperability profiles as defined in the epSOS project and further evolved in the openNCP community. By this the CREDENTIAL eHealth pilot may be operated cross-border by integrating the XDS Façade into a National Contact Point gateway.

B.2.1 IHE XDS Mapping of CREDENTIAL Actors and Services

The following figure sketches the actors (logical building blocks) and transactions (services) as defined by IHE XDS.

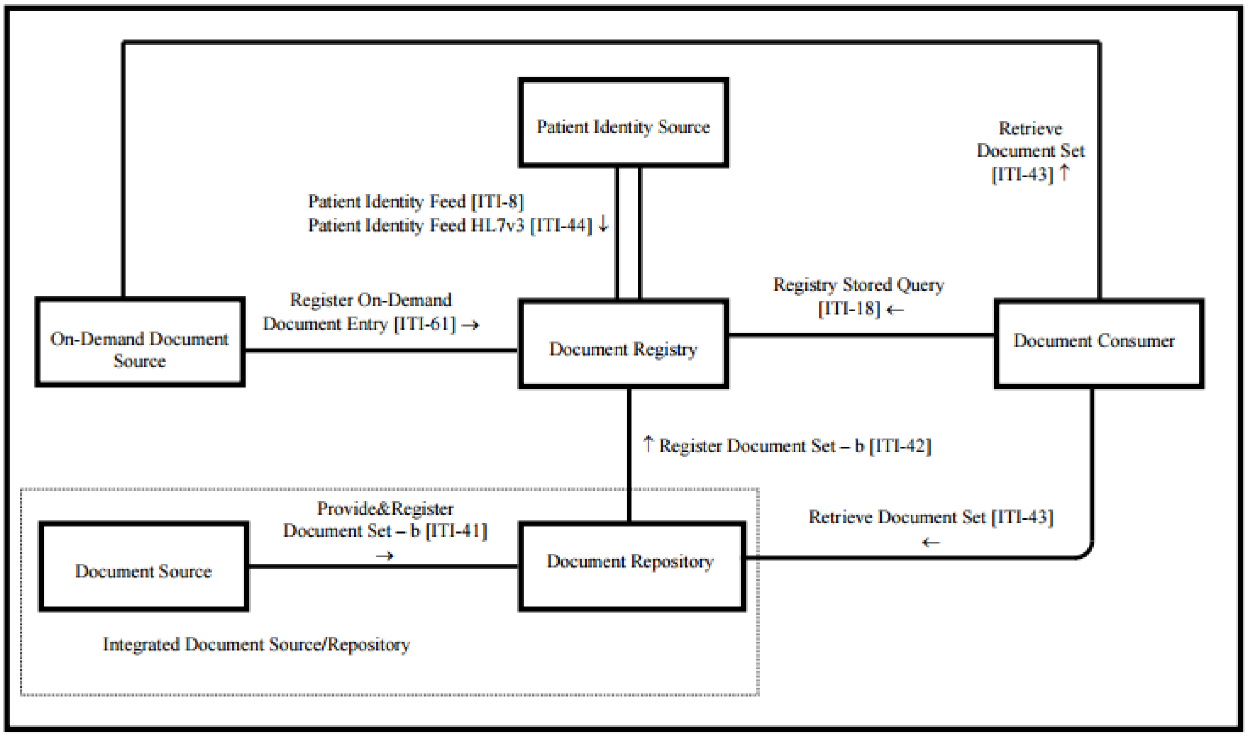

Patient Identity Source
Retrieve Document Set [ITI-43] ↑
Patient Identity Feed [ITI-8]
Patient Identity Feed HL7v3 [ITI-44] ↓
Registry Stored Query [ITI-18] ←
Register On-Demand Document Entry [ITI-61] →
On-Demand Document Source
Document Registry
Document Consumer
↑ Register Document Set – b [ITI-42]
Provide&Register Document Set – b [ITI-41] →
Retrieve Document Set [ITI-43] ←
Document Source
Document Repository
Integrated Document Source/Repository

Figure 32: IHE XDS Actors and Transactions (from IHE ITI TF-1)

IHE XDS implements the ebXML separation of a document registry and a document repository while integrating these with further building blocks:

- The Document Registry takes responsibility for the management of document metadata and provides services for document query and registration. For CREDENTIAL an existing XDS Document Registry implementation will be used.

- The Document Repository implements a content-agnostic store for medical documents. It provides services for storing documents (which will then be registered at the Document Registry by the Document Repository) and for retrieving a set of identified documents. For the CREDENTIAL eHealth pilot, the Document Repository will only store encrypted documents while the respective keys are stored in the CREDENTIAL Wallet.
- The Document Source and On-Demand Document Source actors are the origin of medical documents that are shared through IHE XDS. In CREDENTIAL these actors will be implemented through the patients' mobile devices and the doctors' IT systems.
- The Patient Identity Source takes responsibility for announcing new patient identifiers to the XDS Document Registry (e.g., when a patient is admitted to a hospital). For the CREDENTIAL eHealth pilot this actor will be implemented through the CREDENTIAL identity services.
- The Document Consumer actor is implemented by any component that utilizes document sharing services of the Document Repository and Document Registry. In CREDENTIAL this actor will be implemented through the patients' mobile devices and the doctors' IT systems.

A further actor – Document Administrator – is defined in the IHE Metadata Update integration profile which extends IHE XDS by transactions for updating document metadata:

- The Document Administrator actor may update document metadata which is, e.g., required for deprecating a document.

The figure below shows how the logical actors and services defined for the CREDENTIAL eHealth pilot will be mapped on to IHE actors and transactions.

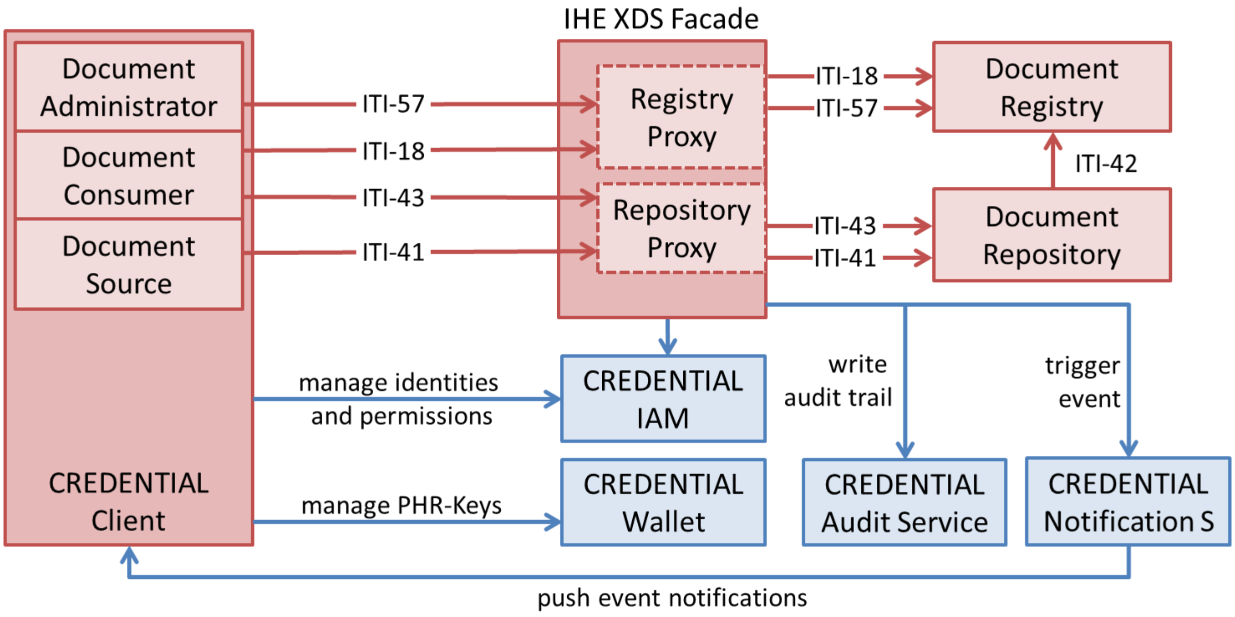


Figure 33: Mapping CREDENTIAL Services onto IHE XDS Transactions

In order to reuse an existing implementation of the IHE actors without any modifications, a façade is introduced as an interceptor which handles all security related interactions with the CREDENTIAL services. The table below summarizes how the façade brokers request messages to IHE actors and CREDENTIAL services.

CREDENTIAL Service	Client -> Façade	Façade -> IHE Actors and CREDENTIAL Services
queryDocuments	ITI-18: Registry Stored Query	Façade -> Document Registry: ITI-18: Registry Stored Query Façade -> IAM Services: Authorize User Façade -> Audit Service: Write Audit Trail Entry
retrieveDocuments	ITI-43: Retrieve Document Set	Façade -> Document Repository: ITI-43: Retrieve Document Set Façade -> IAM Services: Authorize User Façade -> Audit Service: Write Audit Trail Entry Façade -> Notification Service: Trigger event
createDocuments	ITI-41: Provide&Register Document Set	Façade -> Document Repository: ITI-41: Provide&Register Document Set Document Repository -> Document Registry ITI-42: Register Document Set Façade -> IAM Services: Authorize User Façade -> Audit Service: Write Audit Trail Entry Façade -> Notification Service: Trigger event
deprecateDocument	ITI-57: Update Document Set	Façade -> Document Registry: ITI-57: Update Document Set Façade -> IAM Services: Authorize User Façade -> Audit Service: Write Audit Trail Entry Façade -> Notification Service: Trigger event

B.2.2 Metadata Definitions

B.2.2.1 IHE XDS Registry Information Model

The information model implemented by IHE XDS consists of five major kinds of entities:

- DocumentEntry ebXML registry objects capsule the metadata associated to a medical document. These metadata reflect the author, affected patient and creation context of the document together with some technical information such as the document file size and a hash value.
- Each DocumentEntry refers to a medical document that is treated by IHE XDS as a BLOB.
- Medical data is provided to a Document Repository as a SubmissionSet registry object which capsules a set of medical documents together with their belonging DocumentEntry objects. SubmissionSets have metadata of their own and cannot be modified after they have been uploaded to the Document Registry.

- Folder registry objects allow bracing a set of DocumentEntry objects. In XDS folders cannot be nested. Nevertheless any DocumentEntry can be a member to multiple folders. As DocumentEntries, Folder objects can only be registered to the Document Registry as part of a SubmissionSet.
- Relationships among objects are always explicit. They are expressed through Association registry objects that link one object to another. The most important kind of relationship is the hasMember-association which places DocumentEntries into SubmissionSets and/or Folders.

The figure below sketches which associations need to be explicitly defined to upload a single DocumentEntry within a single Folder into a single SubmissionSet. For making the lifecycle of documents explicit, documentEntries may be linked to other document entries using further associations for document replacement, document transformation and document addendum.

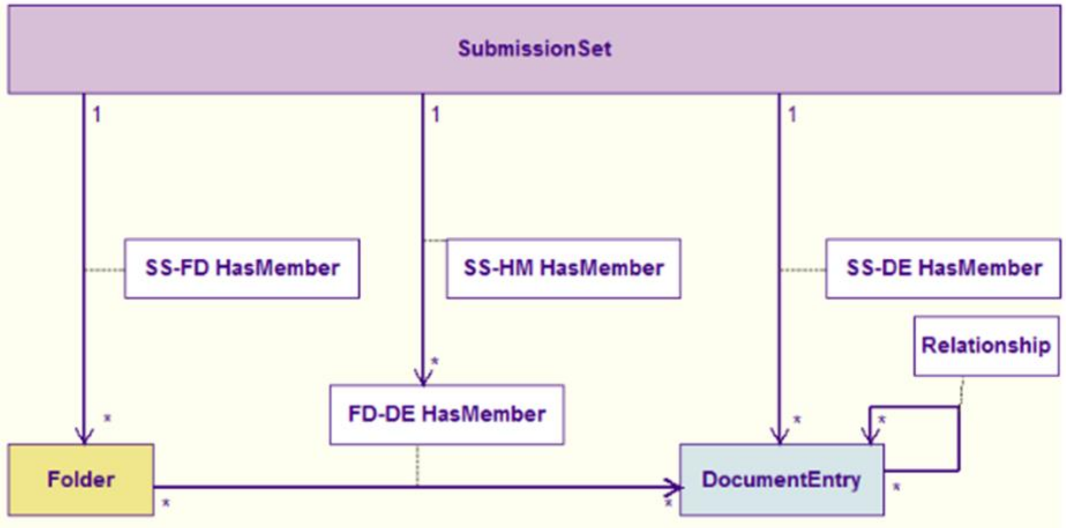


Figure 34: IHE XDS Registry Objects [IHE ITI TF-3]

B.2.2.2 CREDENTIAL Adaptation of the XDS Information Model

As CREDENTIAL only utilizes a subset of the full IHE XDS functionality, a simplified, constrained subset of the IHE XDS information model will be used for the CREDENTIAL eHealth pilot.

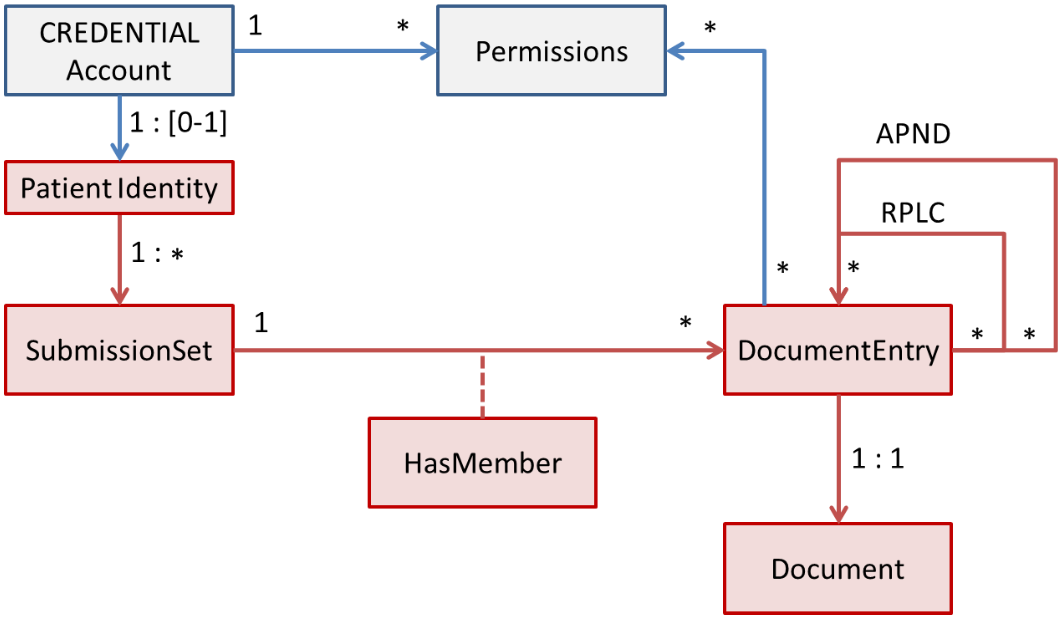


Figure 35: Information Model for the CREDENTIAL eHealth Pilot

The following constraints and simplifications will be applied for CREDENTIAL:

- The Patient Identity associated with a Personal Health Record univocally corresponds to a CREDENTIAL Account. The CREDENTIAL Account ID is used as the unique patient required by IHE XDS. By this there shall not be more than one patient identity linked to a single CREDENTIAL Account.
- XDS Folders SHALL NOT be used. Any attempt to create or access an XDS Folder SHALL result in an error.
- As document relationships only “Append” and “Replace” SHALL be considered valid associations. Any other associations SHALL NOT be accepted by the Document Registry.
- Documents are unique, that is the same document MUST NOT be linked to multiple DocumentEntry objects. If the Document Registry receives a registration request for an already registered document, it SHALL respond with an error.
- Permissions can be bound to patient identities and document entries. Permissions bound to patient identities are linked to the corresponding CREDENTIAL account for handling respective authorization decision requests solely within CREDENTIAL. Permissions on documents are encoded within XACML polices created by the Patient App. These policies are transparent to CREDENTIAL even though the CREDENTIAL PDP Service shall be able to process them based on policy information that is provided by the PEP within the PHR. If no permissions are defined for a CREDENTIAL Account, only the owner of the CREDENTIAL Account (the patient) is granted access permissions to the documents which are linked with this account’s PHR.
- CREDENTIAL MAY define security policies that restrict access to certain types of documents to certain user roles (e. g. there MAY be a policy that clinical documents are only accessible to doctors). These policies are transparent to the PHR and solely handled within the CREDENTIAL IAM services.

B.2.2.3 CREDENTIAL SubmissionSet Metadata Profile

Each SubmissionSet is described by a standard set of metadata attributes. The table below lists all IHE-defined SubmissionSet metadata attributes together with the constraints which SHALL be considered for the CREDENTIAL eHealth pilot. A recommendation on the use of coded values for the defined CREDENTIAL eHealth storyboard is given below. All code systems, value sets and codes referenced in the metadata definitions below are available online at http://semantik.fokus.fraunhofer.de/WebCts2LE/main3/terminologies.jsp.

Attribute	Opt.	CREDENTIAL Constraints
author	1..*	The humans and/or machines that authored the document in the SubmissionSet. Per IHE this attribute contains the sub-attributes: authorInstitution, authorPerson, authorRole, authorSpecialty. For CREDENTIAL the following constraints apply: - If the patient or a patient controlled device is the origin of the data in the SubmissionSet, only the CREDENTIAL patient pseudonym SHALL be given as the ID of the author. The identification scheme shall be *credential-patient-pseudonym*. Sub-elements for names and contact data shall not be used. - If a health professional is the author of the data in the submission set then the name and ID of the responsible healthcare professional organization shall be given for the authorInstitution sub-attribute. For identifying the organization only normative national identification schemes shall be used (e.g., GDA Index in Austria). o If the authorRole sub-attribute is used, only values from the value set *credential-author-roles* shall be used. o Other sub-attributes as defined in [IHE ITI TF-3] may be used but will not be processed or interpreted by CREDENTIAL.
availabilityStatus	1..1	Shall be used according to [IHE ITI TF-3#4.2.3.3.2]. The allowed values are summarized in the CREDENTIAL Availability Status terminology.
comments	0..1	May be provided but will not be processed or interpreted by CREDENTIAL. Comments shall not contain any information that may disclose information about the identity of the patient.
contentTypeCode	1..1	This attribute shall contain a document class code that reflects the overall characteristics of the document package. In most situations, this will be the class code value of the "leading" document in the SubmissionSet. The value of this attribute shall be taken from the CREDENTIAL Document Class Codes value set.
entryUUID	1..1	Shall be used according to [IHE ITI TF-3#4.2.3.3.5].
homeCommunityId	0..1	If provided, the OID of a CREDENTIAL PHR domain shall be given as an URN. The root for this OID shall be *credential-affinity-domains*.
intendedRecipient	0..*	Should not be used as most of the data uploaded to the PHR are targeted at the whole care team (which may evolve over time). If a value is defined for this element, access will only be grated to the document's author, the named intended recipient and the patient.
patientId	1..1	Shall be the CREDENTIAL assigned pseudonym ID of the patient. The identification scheme shall be *credential-patient-pseudonym*.

sourceId	1..1	Shall provide the OID of the organization that is responsible for placing the SubmissionSet into to PHR. If the organization is not assigned an OID from its responsible national authority (e.g., DIMDI in Germany) it shall apply for a project-specific OID under the root-OID *credential-participating-organizations*.
submissionTime	1..1	Shall be used according to [IHE ITI TF-3#4.2.3.3.10].
title	0..1	Should not be used. If provided, the SubmissionSet title should reflect the display name of the *contentTypeCode*.
uniqueId	1..1	Shall be used according to [IHE ITI TF-3#4.2.3.3.12].

B.2.3 Recommendations for the CREDENTIAL eHealth Storyboard

The table below lists the *authorRole* and *contentTypeCode* coded values that should be used for implementing the CREDENTIAL eHealth pilot story board. The values given for *authorRole* must be resolved from the CREDENTIAL value set *CREDENTIAL Author Roles*. The values given for *contentTypeCode* must be resolved from the CREDENTIAL value set *CREDENTIAL Document Class Codes*.

Storyboard	authorRole	contentTypeCode
The diabetologist registers a documentation of the given consent with the PHR.	CAR Doctor	CDCC Patient Consent
The diabetologist uploads the gathered medical status data and the care goal definition to the PHR.	CAR Doctor	CDCC Care Plan
Data from the patient’s personal health devices (scale, glucometer) is regularly stored to the PHR.	CAR PHR	CDCC Protocol
Regularly the diabetologist will add lab reports to the PHR.	CAR Doctor	CDCC Laboratory
The diabetologist assembles care status reports from the available data and uploads these to the PHR.	CAR Doctor	CDCC Assessment
In advance to visits at other specialist doctors the diabetologist assembles relevant data into a short report that is stored in the patient's PHR.	CAR Doctor	CDCC Order
The patient’s activity belt uploads aggregated activity data to the PHR.	CAR PHR	CDCC Protocol
The patient records all food intake and provides the list to the diet specialist via the PHR.	CAR Patient	CDCC Patient Note
Automatically rendered reports about activity and vitals are placed in the PHR every week.	CAR Doctor	CDCC Assessment
The family doctor manages a medication plan for the patient. The medication plan is made available to other doctors through the PHR.	CAR Doctor	CDCC Medication

B.2.3.1 CREDENTIAL DocumentEntry Metadata Profile

Each DocumentEntry is described by a standard set of metadata attributes. The table below lists all IHE-defined DocumentEntry metadata attributes together with the constraints which SHALL be considered for the CREDENTIAL eHealth pilot. A recommendation on the use of coded values for the defined

CREDENTIAL eHealth storyboard is given in Section B.2.4. All code systems, value sets and codes referenced in the metadata definitions below are available online at http://semantik.fokus.fraunhofer.de/WebCts2LE/main3/terminologies.jsp.

Attribute	Opt.	CREDENTIAL Constraints
author	1..*	The humans and/or machines that authored the document. Per IHE this attribute contains the sub-attributes: authorInstitution, authorPerson, authorRole, authorSpecialty. For CREDENTIAL the following constraints apply: - If the patient is the author of the document, only the CREDENTIAL patient pseudonym SHALL be given as the ID of the author. The identification scheme shall be *credential-patient-pseudonym*. Sub-elements for names and contact data shall not be used. The authorRole shall be set to *CAR_Patient*. - If the document was assembled by a patient controlled personal health device, this device shall be identified instead of the patient. The sub-element assignedAuthoringDevice shall be provided. The authorRole shall be set to *CAR_PHD*. - If a health professional or a device/system controlled by the health professional is the author of the document then the name and ID of the authoring person and the responsible healthcare professional organization shall be given. For identifying the author, this person's CREDENTIAL Account identifier shall be used. o If the authorRole sub-attribute is used, only values from the value set credential-author-roles shall be used. o Other sub-attributes as defined in [IHE ITI TF-3] may be used but will not be processed or interpreted by CREDENTIAL.
availabilityStatus	1..1	Shall be used according to [IHE ITI TF-3#4.2.3.2.2]. The allowed values are summarized in the CREDENTIAL Availability Status terminology.
classCode	1..1	This attribute shall contain coarse grained classification of the kind of document. The value of this attribute shall be taken from the *CREDENTIAL Document Class Codes* value set.
comments	0..1	Should not be used. Will not be processed or interpreted by CREDENTIAL.
confidentialityCode	1..1	Only values from the CREDENTIAL value set *CREDENTIAL Confidentiality Codes* shall be used. A value of "R" (restricted) should only be used for documents that shall not be accessible to other persons than the patient and his family doctor.
creationTime	1..1	Shall be used according to [IHE ITI TF-3#4.2.3.2.6].
entryUUID	1..1	Shall be used according to [IHE ITI TF-3#4.2.3.2.7]
eventCodeList	0..*	Should not be used for CREDENTIAL. Event codes provided through this attribute will not be processed or interpreted by CREDENTIAL.
formatCode	1..1	Per IHE formatCodes should be defined by the affinity domain. As

		all document types defined for CREDENTIAL can be distinguished by their typeCode and mimeType there is no need for redundant formatCodes in CREDENTIAL. Therefore the formatCode shall always be set to urn:ihe-d:mime.
hash	0..1	Should be provided for each document. As defined in [IHE DEN] the hash must be calculated over the encrypted document.
healthcareFacilityTypeCode	1..1	Only values from the CREDENTIAL value set *CREDENTIAL Healthcare Facility Types* shall be used.
homeCommunityId	0..1	If provided, the OID of a CREDENTIAL PHR domain shall be given as an URN. The root for this OID shall be *credential-affinity-domains*.
languageCode	1..1	Shall be used according to IHE ITI TF-3#4.2.3.2.13.
legalAuthenticator	0..1	For consent documents this element shall record the person and organization that validated the patient's consent and registered it with CREDENTIAL. For all other documents this attribute should not be used. The restrictions listed for the author attribute apply.
mimeType	1..1	Shall be used according to IHE ITI TF-3#4.2.3.2.15.
patientId	1..1	Shall be the CREDENTIAL assigned pseudonym ID of the patient. The identification scheme shall be *credential-patient-pseudonym*.
practiceSettingCode	1..1	Only values from the CREDENTIAL value set *CREDENTIAL Practice Setting Codes* shall be used. If the patient or a patient controlled device is the origin of the document, the code *CPSC_Not_Applicable* shall be used.
repositoryUniqueId	0..1	Shall not be used for uploading documents. Shall be set by the Document Repository when registering a document with the Document Registry.
serviceStartTime	1..1	Shall be used according to [IHE ITI TF-3#4.2.3.2.19].
serviceStopTime	1..1	Shall be used according to [IHE ITI TF-3#4.2.3.2.20].
size	0..1	Should be provided for all uploaded documents. Shall be used according to [IHE ITI TF-3#4.2.3.2.21].
sourcePatientId	1..1	No value shall be provided for this attribute in order to mask the identity of the patient (IHE XDS allows for a value list with zero elements).
sourcePatientInfo	1..1	No value shall be provided for this attribute in order to mask the identity of the patient (IHE XDS allows for a value list with zero elements).
title	0..1	Should not be used in CREDENTIAL. Must not be set automatically to ensure that no protected information about the patient’s identity is disclosed.
typeCode	1..1	This attribute shall contain fine grained classification of the kind of document. The value of this attribute shall be taken from the *CREDENTIAL Document Type Codes* value set.
uniqueId	1..1	Shall be used according to [IHE ITI TF-3#4.2.3.2.25].
URI	0..1	Should not be used for CREDENTIAL. Must not be set automatically to ensure that no protected information about the patient’s identity is disclosed.
referenceIdList	0..1	Should not be used for CREDENTIAL. IDs provided through this attribute will not be processed or interpreted by CREDENTIAL.

B.2.4 Recommendations for the CREDENTIAL eHealth Storyboard

The table below lists the *classCode* and *typeCode* coded values that should be used for implementing the CREDENTIAL eHealth pilot story board. The values given for *typeCodee* must be resolved from the CREDENTIAL value set *CREDENTIAL Document Type Codes*. The values given for *classCode* must be resolved from the CREDENTIAL value set *CREDENTIAL Document Class Codes*.

Document	classCode	typeCode
(documentation of the) given consent	CDCC Patient Consent	CDTC Basic Consent
medical status report	CDCC Assessment	CDTC Status Report
care goal definition	CDCC Care Plan	CDTC Care Goal Definition
care plan	CDCC Care Plan	CDTC Diabetes Care Plan
monitoring data (from Personal Health Devices)	CDCC Protocol	CDTC PHD Monitoring Data
lab reports	CDCC Laboratory	CDTC Laboratory
diet plan	CDCC Care Plan	CDTC Nutrition Plan
nutrition diary	CDCC Patient Note	CDTC Patient Diary
reports about activity and vitals	CDCC Assessment	CDTC Non-Clinical Summary
medication plan	CDCC Medication	CDTC Medication Plan

The following table gives recommentation for encoding the actor roles as referenced in the CREDENTIAL eHealth storyboard.

Acting role	healthcareFacilityTypeCode	practiceSettingCode
Patient	CHFT Patient	CPSC Not Applicable
Personal Health Device	CHFT Patient	CPSC Not Applicable
Care manager (Diabetologist)	CHFT Practice, CHFT MVZ or CHFT Hospital	CPSC Endocrinology Diabetology
Family Doctor	CHFT Practice	CPSC General Practitioner
Specialist Doctor	CHFT Practice, CHFT MVZ or CHFT Hospital	See ihede-codesystem-4

B.2.5 IHE XDS Bindings for CREDENTIAL Services

B.2.5.1 Create Documents

Providing a document to a CREDENTIAL PHR is bound to the IHE *Provide and Register Document Set* transaction (ITI-41). The CREDENTIAL logical actors map onto the defined IHE technical actors as follows:

CREDENTIAL Logical Actor	IHE XDS Technical Actor (see [IHE ITI TF-2b#3.41.4])
CREDENTIAL Patient App	Content Sender

CREDENTIAL Doctor App	
XDS Façade (PHR)	Content Receiver

B.2.5.2 Request Message

The table below summarizes how the service's logical arguments shall be mapped onto elements of an IHE ITI-41 request message.

Logical Argument	Binding to IHE ITI-41 Request Message (see [IHE ITI TF-2b#3.41.4.1])
accountID	submissionSet/patientID and documentEntry/patientID
document[1..*]	base64 encoded document encrypted with PHR-Key and placed into the request message via MTOM/XOP as defined in [IHE ITI TF-2b]
docMetadata[1..*]	documentEntry
docRelationship[0..*]	ebRIM association objects as defined in [IHE ITI TF-3]

For the CREDENTIAL createDocuments service the following constraints to the IHE ITI-41 request message semantics as defined in [IHE ITI TF-2b#3.41.4.1.2] apply:

- All provided documentEntry objects must be linked to a single submissionSet.
- Metadata for submissionSet and documentEntry objects must follow the definitions given in Section B.2.2.3 of this specification.
- All provided documents must be encrypted according to the IHE Document Encryption Option as defined in Section 5.3 of the IHE Document Encryption (DEN) Integration profile.
- IHE XDS Document Replacement Option and Document Addendum Option shall be supported by both the Content Sender and Content Receiver.
- IHE XDS Document Transformation Option must not be supported. Respective requests must throw an error indicating an invalid message syntax.
- IHE XDS Folder Management Option must not be supported. Respective requests must throw an error indicating an invalid message syntax.
- IHE XDS Asynchronous Web Services Exchange Option must not be supported. Respective requests must throw an error indicating an invalid message syntax.
- IHE Basic Patient Privacy Enforcement Option must not be supported. All requests shall be validated solely through the CREDENTIAL Access Control System.

B.2.6 Expected Actions

The behavior as defined in [IHE ITI TF-2b# 3.41.4.1.3] must be implemented for CREDENTIAL considering the constraints defined above.

B.2.7 Response Message

The createDocument response message shall implement the behavior as defined in [IHE ITI TF-2b#3.41.4.2].

The logical errors defined in the createDocuments Service Functional Model shall be mapped onto IHE error codes as follows:

CREDENTIAL Logical Error Condition	IHE Error Code
The request message utilizes an XDS option that is not supported by CREDENTIAL.	XDSRegistryMetadataError
The requestor is not authorized to upload documents to the identified account.	XDSUnknownPatientId
The identified account does not exist.	XDSUnknownPatientId
The validation of the integrity, completeness and/or proper encoding of the provided documents failed.	InvalidDocumentContent
The validation of the integrity, completeness and/or proper encoding of the provided metadata failed.	XDSRegistryMetadataError
A document update requests refers to a document that does not exist.	PartialReplaceContentNotProcessed
A document addendum request refers to a document that either does not exist or has been deprecated.	PartialAppendContentNotProcessed

B.2.8 Audit Trail Considerations

The Content Sender (CREDENTIAL Client) may write an audit trail entry for the transaction. In this case the audit trail entry must follow the definition at [IHE ITI TF-2b#3.41.5.1.1].

The Content Receiver (XDS Façade) must write an audit trail entry for the transaction as defined in [IHE ITI TF-2b#3.41.5.1.2].

B.2.8.1 Query Documents

Querying for documents (including document relationships) at the CREDENTIAL PHR is bound to the IHE *Registry Stored Query* transaction (ITI-18). The CREDENTIAL logical actors map onto the defined IHE technical actors as follows:

CREDENTIAL Logical Actor	IHE XDS Technical Actor (see [IHE ITI TF-2b#3.18.4])
CREDENTIAL Patient App	Document Consumer

CREDENTIAL Doctor App	
XDS Façade (PHR)	Document Registry

Two separate queries must be performed for discovering document metadata (documentEntry objects) and document relationships. The Document Cosumer actor shall initiate an ITI-18 FindDocuments query followed by an ITI-18 GetAssociations query. If either of these transactions fails the whole logical transaction shall be considered as failed and no document Entry or Association objects shall be returned to the requestor.

B.2.9 Request Message (FindDocuments)

The table below summarizes how the service's logical arguments shall be mapped onto elements of an IHE ITI-18 FindDocuments request message (see [IHE ITI TF-2a#3.18.4.1.2.3.7.1] for details).

Logical Argument	Binding to IHE ITI-41 Request Message (see [IHE ITI TF-2b#3.41.4.1])
accountID	$XDSDocumentEntryPatientId
docClass[0..*]	$XDSDocumentEntryClassCode
docType[0..*]	$XDSDocumentEntryTypeCode
eventTimeSince[0..1]	$XDSDocumentEntryServiceStartTimeFrom

For the CREDENTIAL queryDocuments service the following constraints to the IHE ITI-18 FindDocuments request message semantics as defined in [IHE ITI TF-2a#3.18.4.1.2.3.7.1] apply:

- Document Consumer and Document Provider actors shall support all query parameters as listed in the table above.
- The $XDSDocumentEntryStatus query argument shall always be provided by the Document Consumer. The value of this argument shall always be "urn:oasis:names:tc:ebxml-regrep:StatusType:Approved". The Document Provider must not accept any other document entry status values.
- Document Consumer and Document Provider actors may support further query parameters not listed in the table above. The Document Provider shall respond with an error if the Document Consumer states a query that does contain a query parameter that is not supported by the Document Provider.

B.2.10 Request Message (GetAssociations)

After the successful execution of the ITI-18 FindDocuments transactions the Document Consumer Actor shall immediately trigger an ITI-18 GetAssociations request message (see [IHE ITI TF-2a#3.18.4.1.2.3.7.7] for details). For each documentEntry contained in the response to the FindDocuments transaction a $uuid argument shall be provided in the GetAssociations request message.

For the CREDENTIAL queryDocuments service the following constraints to the IHE ITI-18 getAssociations request message semantics as defined in [IHE ITI TF-2a#3.18.4.1.2.3.7.7] apply:

- The optional $homeCommunityId argument shall not be used. A Document Provider actor shall reject a request that contains a home community ID which does not correspond to its own affinity domain identifier.
- The Document Provider actor shall only respond with association objects that implement the document relationships which are valid to be used for CREDENTIAL (document addendum and document replacement/update).

B.2.11 Expected Actions

The behavior as defined in [IHE ITI TF-2a#3.18.4.1.3] must be implemented for CREDENTIAL considering the constraints defined above.

B.2.12 Response Message

The response message shall implement the behavior as defined in [IHE ITI TF-2a#3.18.4.1.3] considering the response message contents as defined in [IHE ITI TF-2a#3.18.4.1.2.3.7.1] (FindDocuments) and [IHE ITI TF-2a#3.18.4.1.2.3.7.7] (GetAssociations).

The logical errors defined in the queryDocuments Service Functional Model shall be mapped onto IHE error codes as follows:

CREDENTIAL Logical Error Condition	IHE Error Code
The requestor is not authorized to access the identified account.	XDSUnknownPatientId
The identified account does not exist.	XDSUnknownPatientId
The validation of the integrity, completeness and/or proper encoding of the provided query arguments failed.	XDSRegistryMetadataError
There are no documents discovered that match the query and comply with the current user’s permissions.	Empty result set; no error code

B.2.13 Audit Trail Considerations

The Document Consumer (CREDENTIAL Client) may write an audit trail entry for the ITI-18 transactions. In this case the audit trail entry must follow the definition at [IHE ITI TF-2a#3.18.5.1.1].

The Document Provider (XDS Façade) must write an audit trail entry for each of the ITI-18 transactions as defined in [IHE ITI TF-2a#3.18.5.1.2].

B.2.13.1 RetrieveDocument

Downloading selected documents from the CREDENTIAL PHR is bound to the IHE *Retrieve Document Set* transaction (ITI-43). The CREDENTIAL logical actors map onto the defined IHE technical actors as follows:

CREDENTIAL Logical Actor	IHE XDS Technical Actor (see [IHE ITI TF-2b#3.43.4])
CREDENTIAL Patient App CREDENTIAL Doctor App	Document Consumer
XDS Façade (PHR)	Document Repository

B.2.14 Request Message

The table below summarizes how the service's logical arguments shall be mapped onto elements of an IHE ITI-43 Retrieve Document Set request message (see [IHE ITI TF-2a#3.43.4.1] for details).

Logical Argument	Binding to IHE ITI-41 Request Message (see [IHE ITI TF-2b#3.43.4.1])
documentID[1..*]	documentUniqueId

The repositoryUniqueId argument shall be used as defined in [IHE ITI TF-2a#3.43.4.1.2]. A homeCommunityId should not be provided and shall be ignored by the Document Registry.

For the CREDENTIAL retrieveDocuments service the following constraints to the IHE ITI-43 request message semantics as defined in [IHE ITI TF-2a#3.43.4.1.2] apply:

- All requested documents shall be linked with a single CREDENTIAL account.
- Basic Patient Privacy Enforcement Option must not be implemented.

B.2.15 Expected Actions

The behavior as defined in [IHE ITI TF-2a#3.43.4.1.3] must be implemented for CREDENTIAL considering the constraints defined above.

B.2.16 Response Message

The response message shall implement the behavior as defined in [IHE ITI TF-2a#3.43.4.2].

The logical errors defined in the retrieveDocuments Service Functional Model shall be mapped onto IHE error codes as follows:

CREDENTIAL Logical Error Condition	IHE Error Code

The requestor is not authorized to access the account that holds the identified documents.	XDSDocumentUniqueIdError
There are no documents discovered that match the given document IDs and comply with the current user's permissions.	Empty result set; no error code
An identified document does not exist.	Other documents processed; no error code
The integrity check on one or more of the identified documents failed.	XDSRepositoryError
The requested documents are not all linked with the same account.	XDSResultNotSinglePatient

B.2.17 Audit Trail Considerations

The Document Consumer (CREDENTIAL Client) may write an audit trail entry for the ITI-43 transaction. In this case the audit trail entry must follow the definition at [IHE ITI TF-2a#3.43.6.1.1].

The Document Repository (XDS Façade) must write an audit trail entry the ITI-43 transaction as defined in [IHE ITI TF-2a#3.43.6.1.2].

B.2.17.1 DeprecateDocuments

Deprecating selected documents in the CREDENTIAL PHR is bound to the IHE *Update Document Set* transaction (ITI-57). The CREDENTIAL logical actors map onto the defined IHE technical actors as follows:

CREDENTIAL Logical Actor	IHE XDS Technical Actor (see [IHE MDU#3.57.4])
CREDENTIAL Patient App CREDENTIAL Doctor App	Document Administrator
XDS Façade (PHR)	Document Registry

B.2.18 Request Message

For CREDENTIAL only the Update DocumentEntry availabilityStatus type of the ITI-57 transaction shall be used.

The table below summarizes how the service's logical arguments shall be mapped onto elements of an IHE ITI-57 Update Document Set (Update DocumentEntry availabilityStatus) request message (see [IHE MDU#3.57.4.1.3.3.2] for details).

Logical Argument	Binding to IHE ITI-57 Request Message (see [IHE MDU#3.57.4.1.3.3.2])

documentID[1..*]	entryUUID

The logical parameter documentID is bound to the target documentUniqueID indirectly as the logical documentID relates to the DocumentEntry.uniqueID while the IHE transaction requests for the entryUUID of the documentEntry object (document metadata). Therefore, prior to deprecateDocuments, the Document Administrator actor must:

1. obtain the document entry uniqueID (using the queryDocuments service), and
   - use DocumentEntry.entryUUID as the reference to the object that is to be deprecated.

For the CREDENTIAL deprecateDocuments service the following constraints to the IHE ITI-57 request message semantics as defined in [IHE ITI TF-2a#3.57.4.1.3.3.2] apply:

- All documents to be deprecated shall be linked with a single CREDENTIAL account.
- Any other request type than *Update DocumentEntry availabilityStatus* must not be processed. An error must be returned if another request type is given.
- The value of the slot NewStatus in the request message shall be urn:oasis:names:tc:ebxml-regrep:StatusType:Deprecated. Other status values must not be accepted. An error must be returned if another status is requested.

B.2.19 Expected Actions

The behavior as defined in [IHE MDU#3.57.4.1.2] and [IHE MDU#3.57.4.1.3.3.2] must be implemented for CREDENTIAL considering the constraints defined above.

B.2.20 Response Message

The response message shall implement the behavior as defined in [IHE ITI TF-2a#3.57.4.2].

The logical errors defined in the deprecateDocuments Service Functional Model shall be mapped onto IHE error codes as follows:

CREDENTIAL Logical Error Condition	IHE Error Code
The requestor is not authorized to access the account that holds the identified documents.	XDSDocumentUniqueIdError
There are no document entries discovered that match the given UUIDs and comply with the current user's permissions.	XDSDocumentUniqueIdError
An identified document does not exist.	XDSDocumentUniqueIdError
The requested documents are not all linked with the same account.	XDSResultNotSinglePatient

It shall be noted that [IHE DMU] requests the full transaction to fail if the metadata update of a single document failed.

B.2.21 Audit Trail Considerations

The Document Administrator (CREDENTIAL Client) may write an audit trail entry for the ITI-57 transaction. In this case the audit trail entry must follow the definition at [IHE MDU#3.57.4.1.4.1.1].

The Document Registry (XDS Façade) must write an audit trail entry the ITI-57 transaction as defined in [IHE ITI TF-2a#3.57.4.1.4.1.2].

B.2.22 Security Bindings

B.2.22.1 CREDENTIAL Binding to IHE XUA

Each request to the XDS Façade shall contain a SAML Assertion considering to the IHE Cross-Enterprise User Assertion profile (XUA) in its security header. For CREDENTIAL pilot operations, the following constraints to [IHE ITI TF-2b#3.40.4] apply:

- the mandatory element "AudienceRestriction" shall hold an URI that refers to the CREDENTIAL project,
- the IHE XUA Authz-Consent Option shall not be used as it may bypass CREDENTIAL security mechanisms,
- a Patient Identifier Attribute (see [IHE ITI TF# 3.40.4.1.2.2.1]) shall not be provided as CREDENTIAL does not foresee means for mapping local patient IDs onto CREDENTIAL Account IDs.

If the patient himself is the requestor of PHR services, the following additional constraints ("IHE XUA local policy") apply:

- The attribute "urn:oasis:names:tc:xspa:1.0:subject:subject-id" shall not be used as it may disclose the real-world identity of the patient.

If the request is triggered by the CREDENTIAL Doctor App, the epSOS HCP Identity Assertion[3] profile on IHE XUA shall be used for all provided attributes. The following further constraints shall be implemented:

- The identity attribute "XSPA permissions according with Hl7" should not be used. It must be ignored by the service provider (XDS Façade).
- Per default, the "Purpose of Use" should be considered as "TREATMENT".

[3] See https://publicwiki-01.fraunhofer.de/epSOS_specification/index.php/EpSOS_HP_Identity_Assertion_-_SAML_Binding

B.2.22.2 CREDENTIAL Binding to IHE APPC

The IHE Advanced Patient Privacy Consents (APPC) Integration Profile defines how XACML syntax and attribute namespaces shall be used for defining access control policies in the healthcare domain.

For the CREDENTIAL eHealth pilot the XACML polices generated through the "Authorize Doctor" Technical Use Case (see Section B.3.2.2) shall only use the following attributes:

APPC Category	Attribute	Semantics
Subject	User ID	Restrict access to an identified person
Subject	User Organization ID	Restrict access to an identified organization
Subject	User Role	Used with "List Address Book" for identifying user roles (e.g., "Diabetologist")
Resources:General	Patient ID	Restrict access to data linked with an identified CREDENTIAL Account ID. This attribute must be provided within each policy.
Resources:DocumentEntry	Document Type	Only used for granting access to the PHR-Key.
Resources:DocumentEntry	Document Unique ID	Restrict access to an identified document (used for ad hoc authorization only).

Actions shall be coded as URIs as defined in [IHE APPC# 5.6.2.1.6.1].

For the types of permission defined for the CREDENTIAL eHealth pilot, this results in the following definitions:

Permission	Policy
Registered User	- User ID = User's CREDENTIAL Account ID or User Organization ID = User's CREDENTIAL Account ID - Patient ID = Patient's CREDENTIAL Account ID access granted to two objects: 2. Document Type = "PHR Key", action = "read" 3. action = "urn:ihe:iti:2007:ProvideAndRegisterDocumentSet-b"
Role Authorization	- User ID = User's CREDENTIAL Account ID or User Organization ID = User's CREDENTIAL Account ID - UserRole = "Diabetologist" or "Family Doctor" - Patient ID = Patient's CREDENTIAL Account ID access granted to two objects: 1. Document Type = "PHR Key", action = "read" 2. action = "urn:ihe:iti:2007:ProvideAndRegisterDocumentSet-b" or action = "urn:ihe:iti:2007:RegistryStoredQueryResponse" or action = "urn:ihe:iti:2007:RetrieveDocumentSetResponse"
Ad Hoc Authorization	- User ID = User's CREDENTIAL Account ID or User Organization ID = User's CREDENTIAL Account ID - Patient ID = Patient's CREDENTIAL Account ID access granted to two objects: 1. Document Type = "PHR Key", action = "read" 2. Document Unique ID = *Document Identifier* and (action = "urn:ihe:iti:2007:RegistryStoredQueryResponse" or action = "urn:ihe:iti:2007:RetrieveDocumentSetResponse")

B.3 Technical Use Cases for PHR Services interacting with CREDENTIAL

This section specifies all technical use cases that require exchanging messages between PHR services and CREDENTIAL services.

B.3.1 Patient Registration

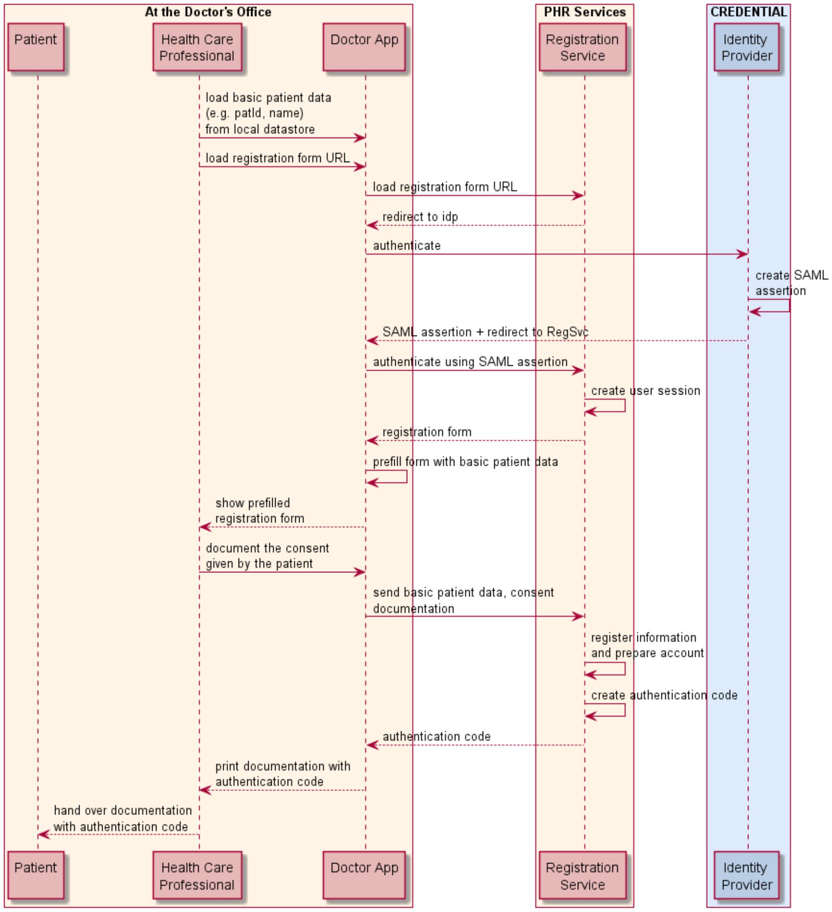
The logical use case “Initialize PHR” (see Section B.1.2) not only registers the patient with CREDENTIAL and the PHR but even establishes a trust relationship between the CREDENTIAL Patient App on the patient’s smartphone and the PHR services. For this the implementation of this use case requires a two-step-approach:

- In the first step the PHR Registration Service issues a secret authentication code that is linked with a registered patient. The registration is performed by the doctor who takes responsibility for the proper identification of the patient.
- In the second step the patient sends the secret authentication code back to the Registration Service together with data that univocally identifies and authenticates his smartphone. This allows the Registration Service to register this smartphone as a trusted device for that patient. In addition, the Registration Service registers the patient’s CREDENTIAL Account ID as the patient ID with the PHR.

In this both steps are specified as separate technical Use Cases.

B.3.1.1 Technical Use Case: Register Patient and Prepare Device Authentication

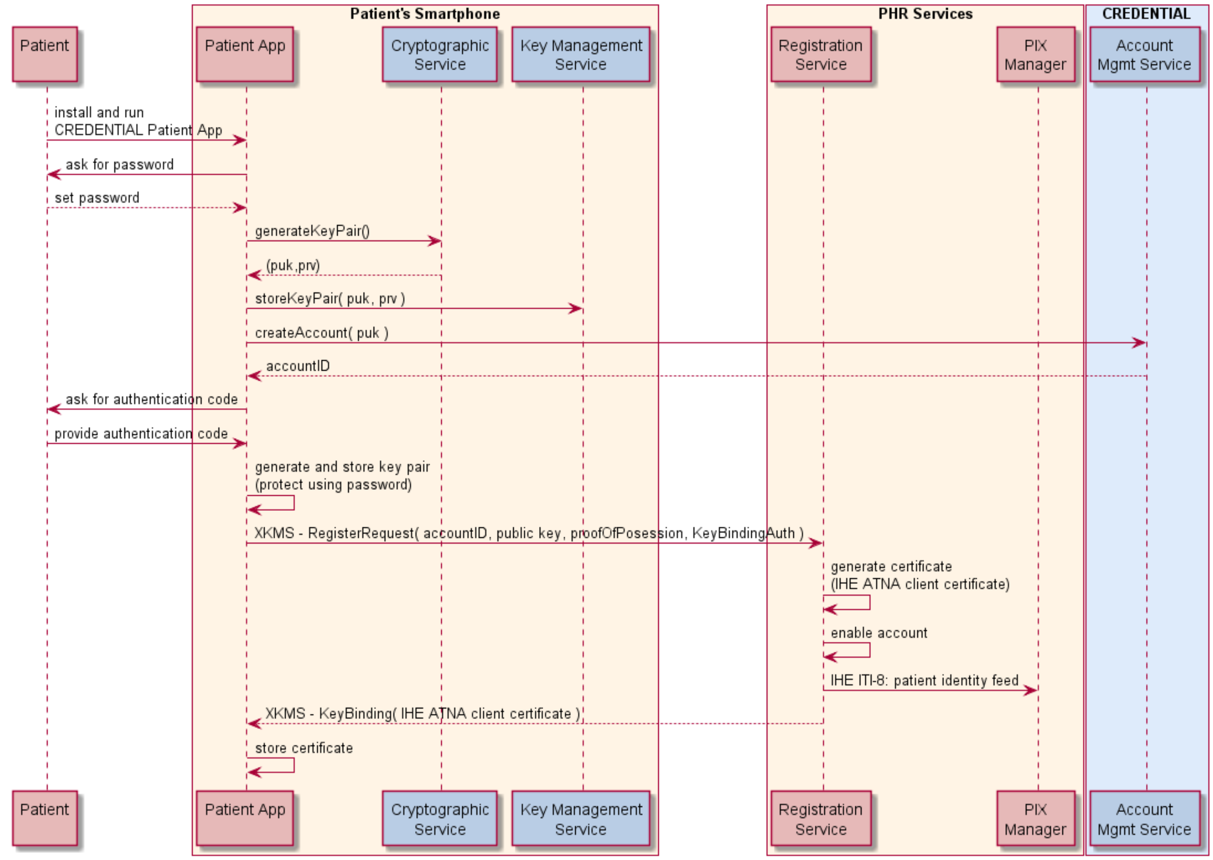
This technical use case refines the logical use cases “Register Patient as eHealth Pilot Participant” (see Section B.1.2.1).

Use Case Name	Register Patient as eHealth Pilot Participant
ID	E.HLT-TUC-011-1
Main Actor	- CREDENTIAL Doctor App
Secondary Actors	- Doctor (Health Care Professional) - Registration Service - Identity Provider - Patient
Pre-conditions	- Patient has given consent to the doctor - Doctor is registered as CREDENTIAL participant
Post-conditions	- PHR is registered with the PHR Registration Service - Patient is prepared for linking his CREDENTIAL account with the PHR - Patient is prepared for registering his smartphone as a trusted device
Description	The doctor registers the patient as a CREDENTIAL eHealth pilot participant with the PHR provider. An authentication code is issued that allows the patient to link the PHR registration with his smartphone and with his CREDENTIAL Account.
Image	

At the Doctor's Office
PHR Services
CREDENTIAL
Patient
Health Care Professional
Doctor App
Registration Service
Identity Provider
load basic patient data (e.g. patId, name) from local datastore
load registration form URL
load registration form URL
redirect to idp
authenticate
create SAML assertion
SAML assertion + redirect to RegSvc
authenticate using SAML assertion
create user session
registration form
prefill form with basic patient data
show prefilled registration form
document the consent given by the patient
send basic patient data, consent documentation
register information and prepare account
create authentication code
authentication code
print documentation with authentication code
hand over documentation with authentication code
Patient
Health Care Professional
Doctor App
Registration Service
Identity Provider

B.3.1.2 Technical Use Case: Link PHR to CREDENTIAL Account

This technical use case refines the logical use cases “Link PHR to CREDENTIAL Account” (see Section 0).

Use Case Name	Link PHR to CREDENTIAL Account
ID	E.HLT-TUC-011-2
Main Actor	CREDENTIAL Patient App
Secondary Actors	- Patient - Registration Service - Identity Provider - PIX Manager
Pre-conditions	- PHR has been initialized for the patient - Patient received the authentication code for one-time-access to his PHR
Post-conditions	- PHR is linked with an CREDENTIAL account - Patient App is installed and ready for use - Patient’s smartphone is prepared to act as a trusted device with respect to the conditions of the IHE ATNA integration profile.
Description	The patient installs the CREDENTIAL Patient App on his smartphone and activates the registration. A new CREDENTIAL account is setup. The patient’s CREDENTIAL account is linked with the patient’s PHR. The patient’s smartphone receives a certificate to establish an IHE ATNA secured communication with the PHR.
Image	 Patient's Smartphone PHR Services CREDENTIAL Patient Patient App Cryptographic Service Key Management Service Registration Service PIX Manager Account Mgmt Service install and run CREDENTIAL Patient App ask for password set password generateKeyPair() (puk,prv) storeKeyPair(puk, prv) createAccount(puk) accountID ask for authentication code provide authentication code generate and store key pair (protect using password) XKMS - RegisterRequest(accountID, public key, proofOfPosession, KeyBindingAuth) generate certificate (IHE ATNA client certificate) enable account IHE ITI-8: patient identity feed XKMS - KeyBinding(IHE ATNA client certificate) store certificate Patient Patient App Cryptographic Service Key Management Service Registration Service PIX Manager Account Mgmt Service

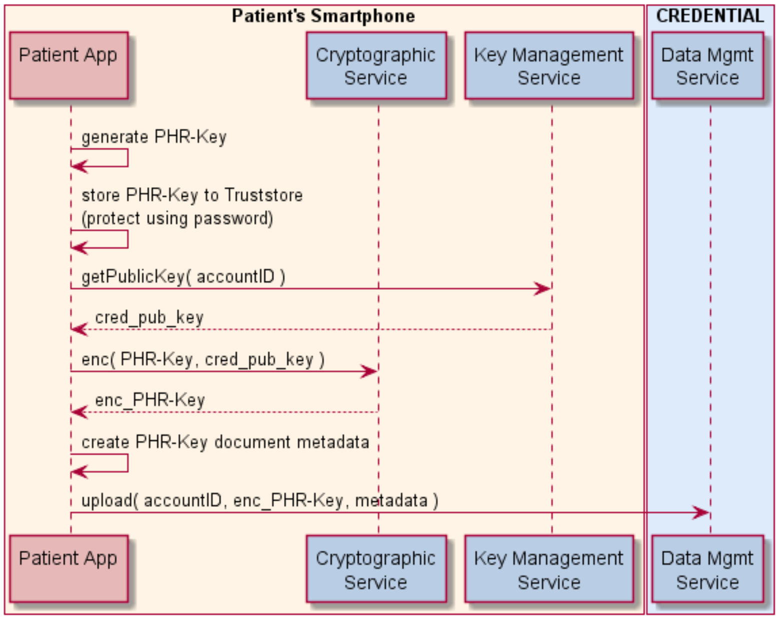
B.3.2 PHR Key Management

For each patient's Personal health Record (PHR) a dedicated PHR-Key is generated and securely stored with the CREDENTIAL Wallet. Using a hybrid encryption approach all documents are encrypted with a dedicated key which is again encrypted with the PHR-Key. Authorized users may gain access to the PHR-Key through the CREDENTIAL proxy re-encryption mechanism.

This section lists the technical use cases for registering and sharing PHR keys. The invalidation of a PHR-Key (e.g., if a patient revoked his consent to CREDENTIAL) is performed through the common CREDENTIAL means for deactivating a user account and therefore will not be further discussed here.

B.3.2.1 Technical Use Case: Create and Register PHR Key

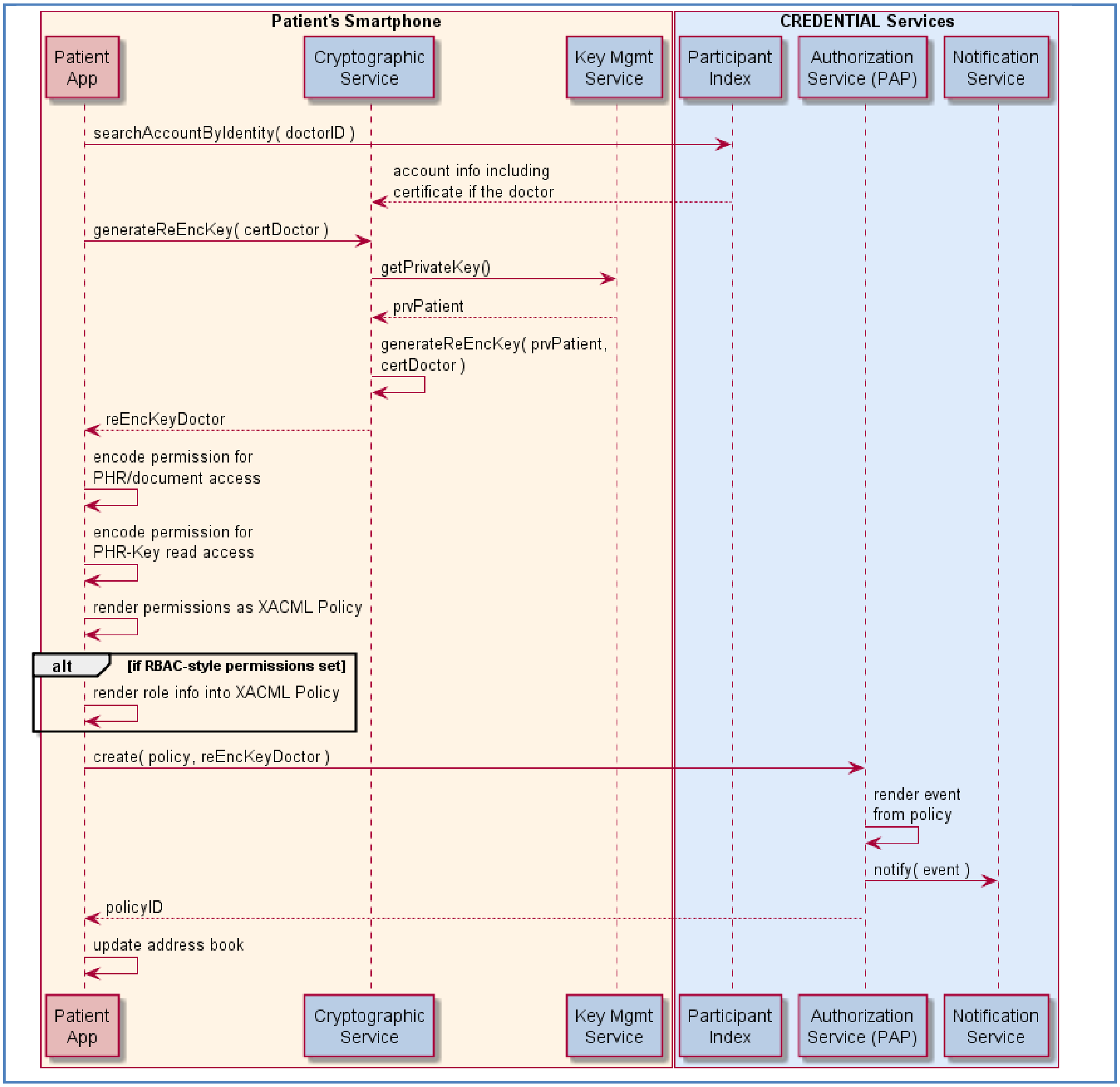
This technical use case refines the logical use cases "Create and Register PHR Key" (see Section B.1.2.3).

Use Case Name	**Create and Register PHR-Key**
ID	E.HLT-LUC-013
Main Actor	CREDENTIAL Patient App
Secondary Actors	- Cryptographic Service - Key Management Service - Data Management Service
Pre-conditions	- Patient has the Patient App installed on his smartphone - Patient is registered as CREDENTIAL participant - Patient has successfully been registered as a PHR participant - Patient is logged in to CREDENTIAL
Post-conditions	- PHR-Key is generated and stored to the CREDENTIAL Wallet
Description	The Patient App generates a PHR-Key for the patient's PHR and stores the key to the CREDENTIAL Data Repository. Authorized users may get access to the PHR-Key through CREDENTIAL's proxy re-encryption service.
Image	

B.3.2.2 Technical Use Case: Authorize Doctor

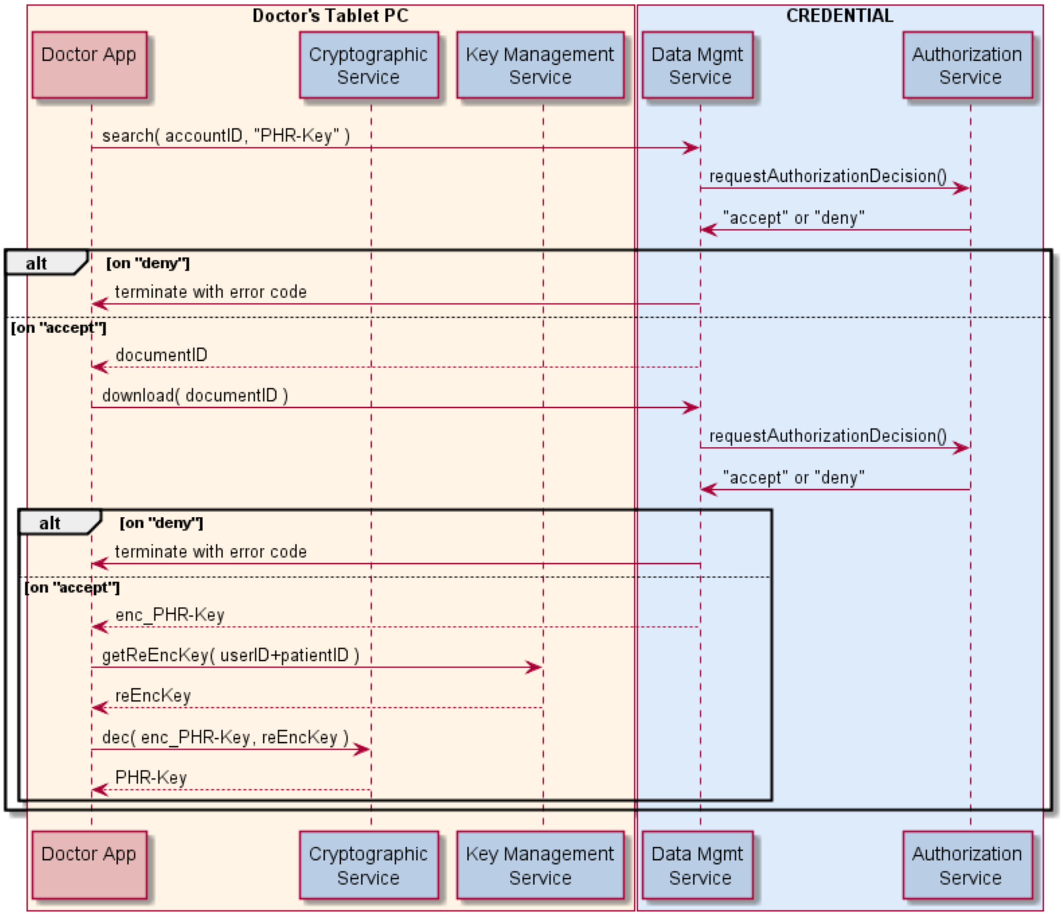
This technical use case refines the logical use cases “Authorize Doctor” (see Section B.1.10.1).

Use Case Name	Authorize Doctor
ID	E.HLT-TUC-091-1
Main Actor	- CREDENTIAL Patient App
Secondary Actors	- Authorization Service - Re-Encryption Key Generation Service
Pre-conditions	- Patient has the CREDENTIAL Patient App installed and activated - Patient is authenticated with the App and with CREDENTIAL - Doctor whose permissions are to be changes is identified - Patient has confirmed the permission changes to be applied
Post-conditions	- New permissions are registered for a doctor. - Re-Encryption key is available for the doctor for accessing the patient’s PHR-Key - The doctor is notified about his permissions
Description	The CREDENTIAL Patient App requests a re-encryption key for the user. It assembles a policy that reflects the granted permissions on the PHR and gives the selected doctor read-access to the PHR-Key in the CREDENTIAL Wallet. By providing the policy and the re-encryption key to the Policy Administration Point the new permissions are activated.
Image	


Patient's Smartphone
CREDENTIAL Services
Patient App
Cryptographic Service
Key Mgmt Service
Participant Index
Authorization Service (PAP)
Notification Service
searchAccountByIdentity(doctorID)
account info including certificate if the doctor
generateReEncKey(certDoctor)
getPrivateKey()
prvPatient
generateReEncKey(prvPatient, certDoctor)
reEncKeyDoctor
encode permission for PHR/document access
encode permission for PHR-Key read access
render permissions as XACML Policy
alt
[if RBAC-style permissions set]
render role info into XACML Policy
create(policy, reEncKeyDoctor)
render event from policy
notify(event)
policyID
update address book

B.3.2.3 Technical Use Case: Obtain PHR Key

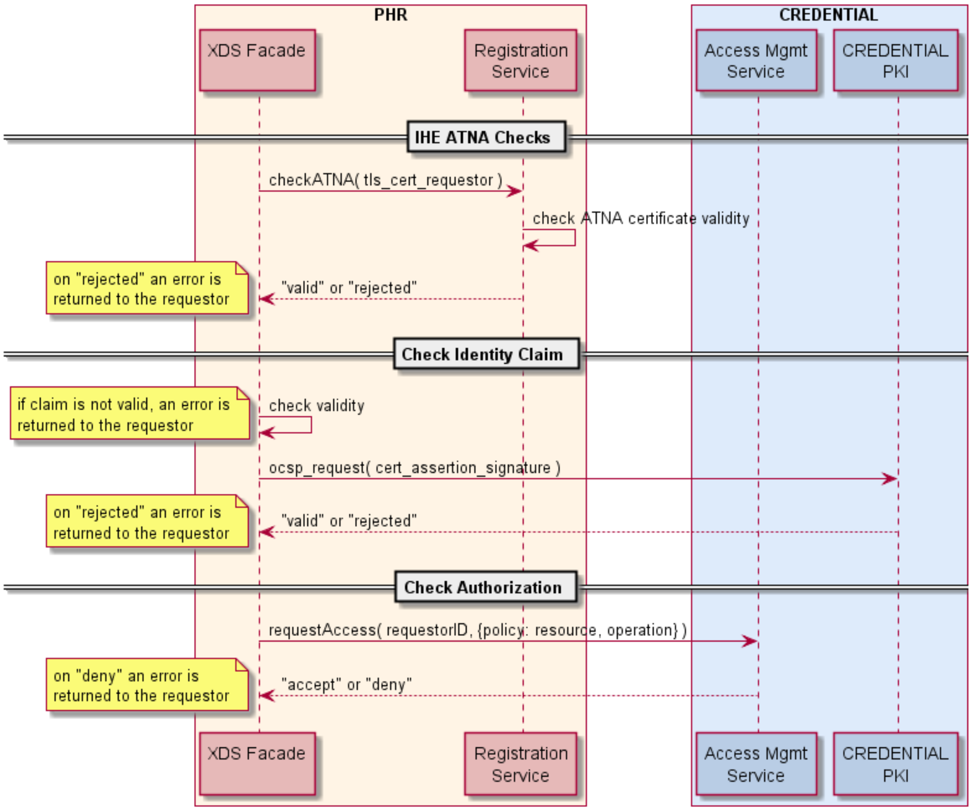
This technical use case refines the logical use cases “Obtain PHR Key” (see Section B.1.9).

Use Case Name	**Obtain PHR-Key**
ID	E.HLT-LUC-013-1
Main Actor	CREDENTIAL Doctor App
Secondary Actors	- Cryptographic Service - Data Management Service
Pre-conditions	- Doctor has the Doctor App installed on his tablet - Doctor is logged in to CREDENTIAL - Patient has a PHR initialized and doctor is authorized to access at least one document within the PHR
Post-conditions	- PHR-Key is available with the Doctor App for decrypting documents obtained from the patient’s PHR
Description	The doctor fetches the encrypted PHR-Key as a document from the CREDENTIAL Wallet and decrypts it using the decryption key that was created for him by the patient.
Image	 Doctor's Tablet PC CREDENTIAL Doctor App Cryptographic Service Key Management Service Data Mgmt Service Authorization Service search(accountID, "PHR-Key") requestAuthorizationDecision() "accept" or "deny" alt [on "deny"] terminate with error code [on "accept"] documentID download(documentID) requestAuthorizationDecision() "accept" or "deny" alt [on "deny"] terminate with error code [on "accept"] enc_PHR-Key getReEncKey(userID+patientID) reEncKey dec(enc_PHR-Key, reEncKey) PHR-Key Doctor App Cryptographic Service Key Management Service Data Mgmt Service Authorization Service

B.3.3 PHR Access Control

B.3.3.1 Technical Use Case: Validate Requestor Authenticity and Authorization

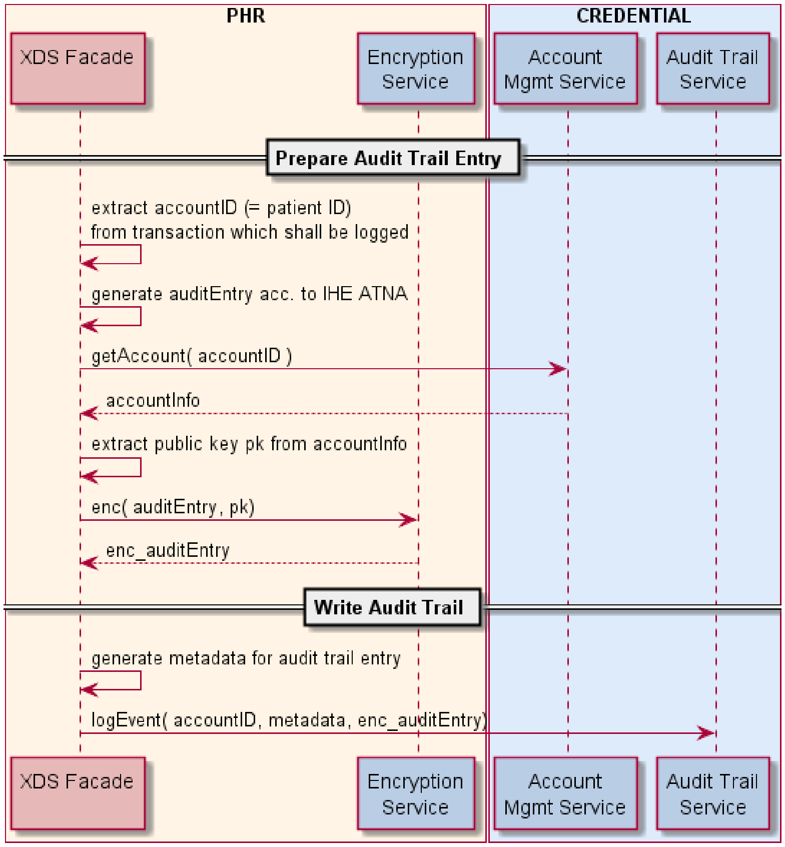
This technical use case refines the logical use case “Validate Requestor Authenticity and Authorization” (see Section B.1.3.2).

Use Case Name	Validate Requestor Authenticity and Authorization
ID	E.HLT-TUC-022-1
Main Actor	- XDS Façade
Secondary Actors	- Authorization Service - CREDENTIAL PKI - Registration Service
Pre-conditions	- A PHR access requests has been accepted by the PHR Façade
Post-conditions	- The access requests is validated with respect to the authenticity of the requestor and the permission of the requestor to access the affected PHR
Description	The XDS Façade validates the authenticity and authorization of the requestor: - Are the requirements for an IHE ATNA secure application connectivity fulfilled? - Is the identity claim authentic and valid? - Has the requestor sufficient permission to perform the requested operation on the identified PHR?
Image	 PHR CREDENTIAL XDS Facade Registration Service Access Mgmt Service CREDENTIAL PKI IHE ATNA Checks checkATNA(tls_cert_requestor) check ATNA certificate validity on "rejected" an error is returned to the requestor "valid" or "rejected" Check Identity Claim if claim is not valid, an error is returned to the requestor check validity ocsp_request(cert_assertion_signature) on "rejected" an error is returned to the requestor "valid" or "rejected" Check Authorization requestAccess(requestorID, {policy: resource, operation}) on "deny" an error is returned to the requestor "accept" or "deny" XDS Facade Registration Service Access Mgmt Service CREDENTIAL PKI

B.3.4 Audit Trail

B.3.4.1 Technical Use Case: Write Audit Trail

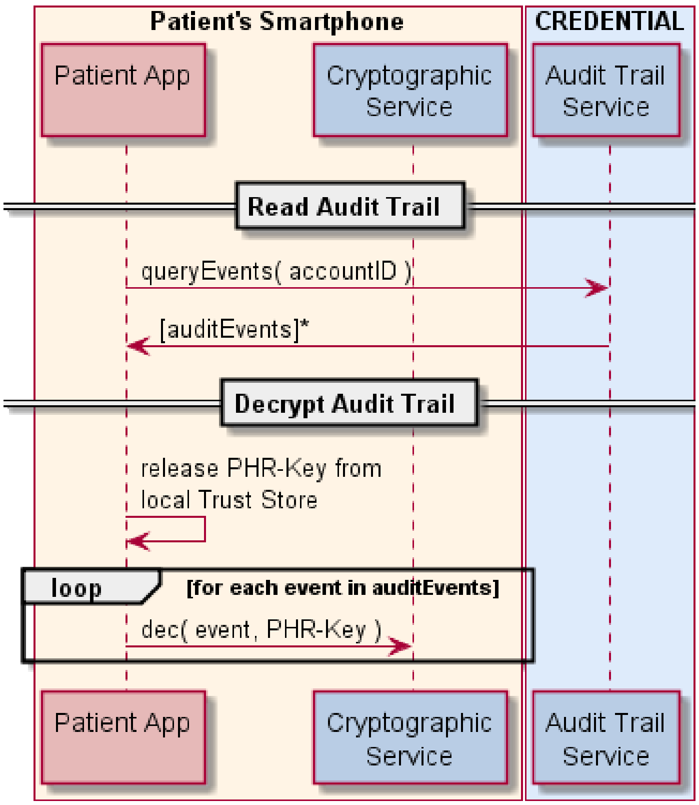
This technical use case refines the logical use case “Write Audit Trail Entry” (see Section B.1.13).

Use Case Name	Write Audit Trail
ID	E.HLT-LUC-014-1
Main Actor	XDS Façade
Secondary Actors	- Cryptographic Service - Key Management Service - Audit Service
Pre-conditions	- An auditable event results from a transaction that was processed by the XDS Facade - The XDS Façade was able to extract the patient ID (equals to the CREDENTIAL Account ID) from that transaction
Post-conditions	- An encrypted audit trail entry is written to the Audit Log
Description	The XDS Façade generates an audit trail entry acc. to IHE ATNA and the CREDENTIAL profiles (see Section B.2.2). It obtains the public key of the patient and encrypts the audit trail entry with that key. The XDS Façade uploads the audit trail entry to the CREDENTIAL Audit Service.
Image	

PHR
CREDENTIAL
XDS Facade
Encryption Service
Account Mgmt Service
Audit Trail Service
Prepare Audit Trail Entry
extract accountID (= patient ID) from transaction which shall be logged
generate auditEntry acc. to IHE ATNA
getAccount(accountID)
accountInfo
extract public key pk from accountInfo
enc(auditEntry, pk)
enc_auditEntry
Write Audit Trail
generate metadata for audit trail entry
logEvent(accountID, metadata, enc_auditEntry)
XDS Facade
Encryption Service
Account Mgmt Service
Audit Trail Service

B.3.4.2 Technical Use Case: Read Audit Trail

This technical use case refines the logical use case “Read Audit Trail Entry” (see Section B.1.14).

Use Case Name	Write Audit Trail
ID	E.HLT-LUC-014-2
Main Actor	XDS Façade
Secondary Actors	- Cryptographic Service - Key Management Service - Audit Service
Pre-conditions	- The patient has the CREDENTIAL Patient App installed and running - The patient is logged in to the App and to CREDENTIAL - The PHR-Key to the patient’s PHR is available in the local trust store
Post-conditions	- Audit log entries on activities on the patient’s PHR data are available for display and/or further processing
Description	
Image	

C eBusiness Use Cases

This section defines the logical and technical use cases for the eBusiness pilot. It starts in Section C.1 with the description of the logical use cases. These logical use cases are further broken down and described in more detail in Section C.2.

C.1 Logical Use Cases

In the following paragraphs are explained the logical Use Cases based on the Business Use Cases described in Work Package 2.

C.1.1 Activate encrypted Legalmail forward

The use case “Activate encrypted Legalmal forward” is part of the selected use case “Legalmail service: Legalmail encrypted message forward” as described in Section 6.4.3.

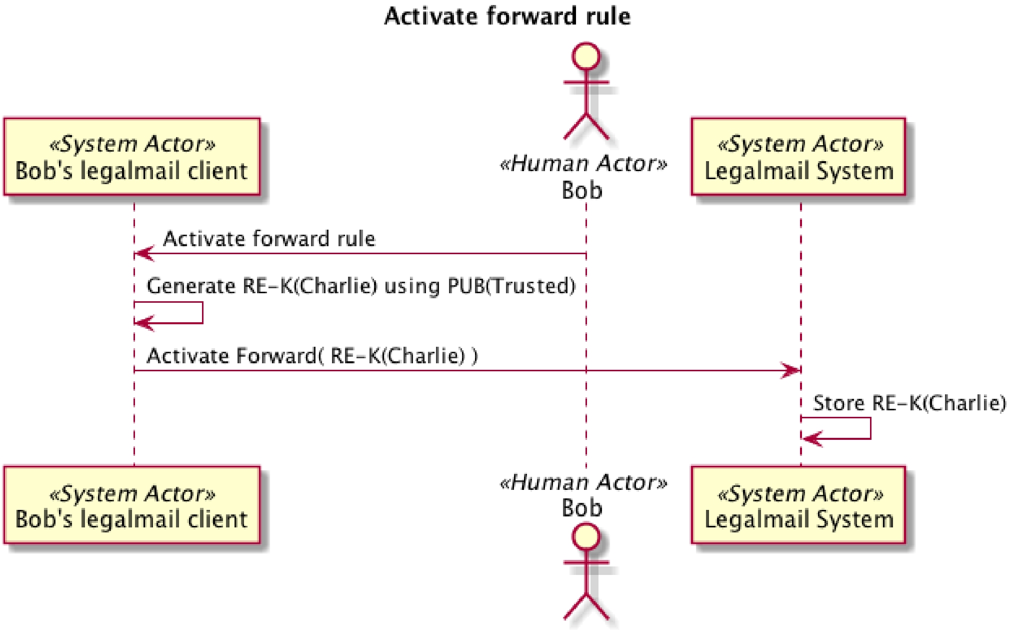
It is composed by the following use cases:

- Activate message forward rule
- Compose encrypted message
- Send encrypted message
- Receive forwarded re-encrypted message

The use cases are explained in the following subsections.

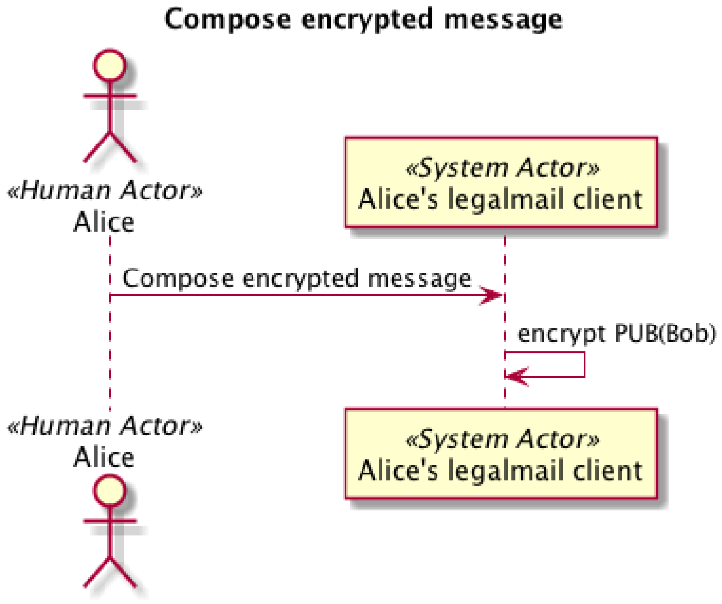
C.1.1.1 Activate message forward rule

This use case explains the means to create a forward rule in the Legalmail System by using the Legalmail Client. By doing so, the user has to activate a forward rule and select the participant who will receive forwarded mails. Therefore, the Legalmail Client creates re-encryption keys based on the given identities. The re-encryption keys are bundled together with the forward rule and submitted to the Legalmail System.

Use Case Name	**Activate message forward rule**
ID	E-BUS-LUC-753
Main Actor	- Legalmail user
Secondary Actors	- Legalmail client - Legalmail service
Pre-conditions	- Legalmail user is using a client that support encryption - Legalmail user have the forward receiver public key
Post-conditions	- The forward rule is configured and the re-encryption key is stored in LegalmailLegalmail system
Description	- Legalmail user Bob chooses to configure a forward rule towards another user
Image	 Activate forward rule «System Actor» Bob's legalmail client «Human Actor» Bob «System Actor» Legalmail System Activate forward rule Generate RE-K(Charlie) using PUB(Trusted) Activate Forward(RE-K(Charlie)) Store RE-K(Charlie) «System Actor» Bob's legalmail client «Human Actor» Bob «System Actor» Legalmail System

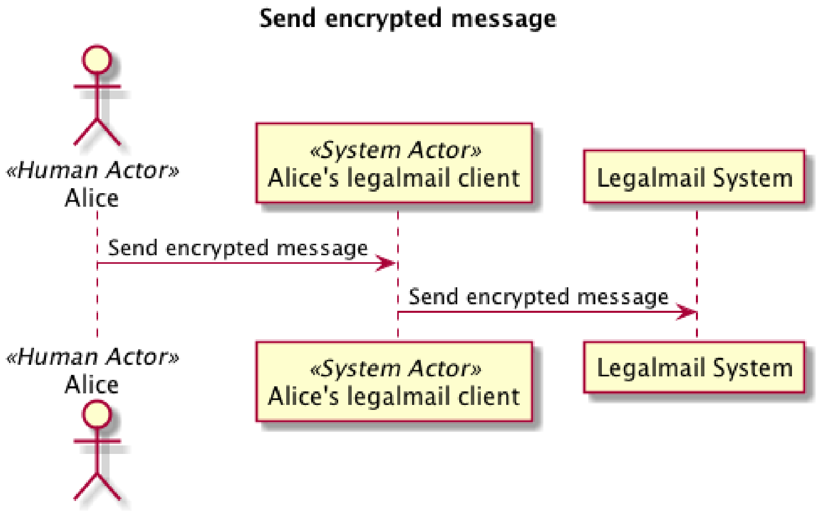
C.1.1.2 Compose encrypted message

This use case explains how mails will be encrypted. Encryption is performed on the user's Legalmail Client by using the public key of the recipient of the mail.

Use Case Name	Compose encrypted message
ID	E-BUS-LUC-751
Main Actor	- Legalmail user
Secondary Actors	- Legalmail client
Pre-conditions	- Legalmail user is using a client that support encryption - Legalmail user have the receiver public key
Post-conditions	- The message is composed and encrypted
Description	- Legalmail user Alice chooses to compose an encrypted message using a well configured Legalmail client
Image	 Compose encrypted message «Human Actor» Alice «System Actor» Alice's legalmail client Compose encrypted message encrypt PUB(Bob) «Human Actor» Alice «System Actor» Alice's legalmail client

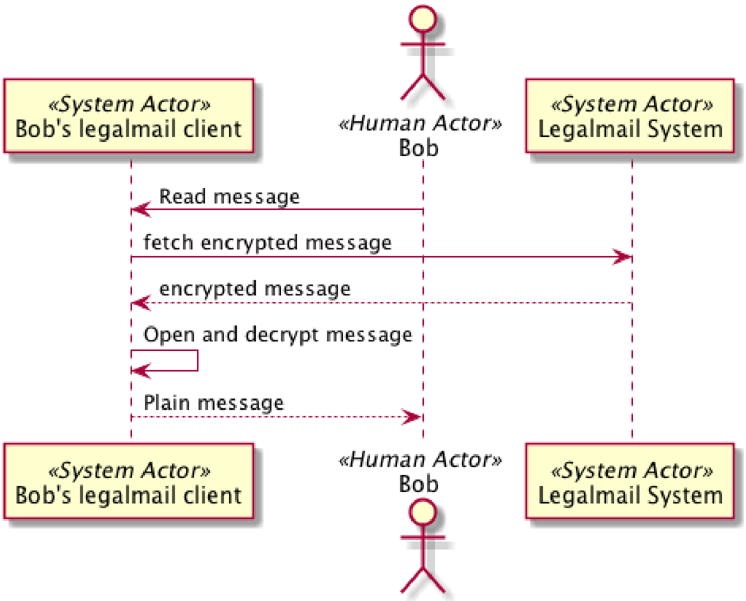
C.1.1.3 Send encrypted message

This use case explains how the encrypted mails are sent to the recipient. By using the Legalmail infrastructure every Legalmail Client sends the encrypted mail to the Legalmail System. From here on the remaining mail delivery process is triggered.

Use Case Name	Send encrypted message
ID	E-BUS-LUC-752
Main Actor	- Legalmail user
Secondary Actors	- Legalmail client - Legalmail service
Pre-conditions	- Legalmail user is using a client that support encryption - Legalmail user have the receiver public key - Legalmail user have composed an encrypted message
Post-conditions	The encrypted message is sent with Legalmail service
Description	- Legalmail user chooses to send a previously composed encrypted message using a well configured Legalmail client
Image	 Send encrypted message «Human Actor» Alice «System Actor» Alice's legalmail client Legalmail System Send encrypted message Send encrypted message «Human Actor» Alice «System Actor» Alice's legalmail client Legalmail System

C.1.1.4 Receive encrypted message

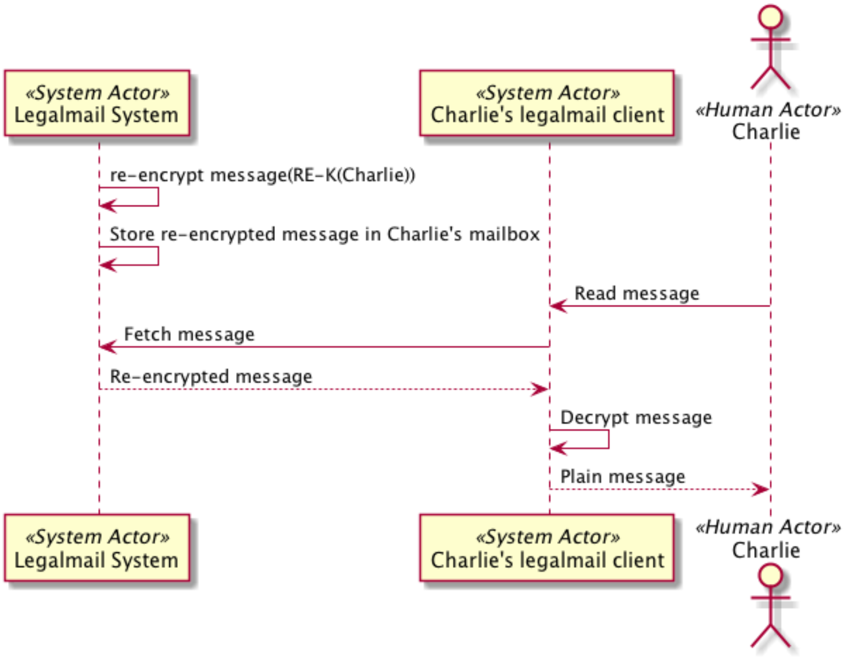
This use case explains how an encrypted mail is received. Two steps have to be performed by the recipient. First the user has to read new mails through the Legalmail Client. The Legalmail Client checks the Legalmail System for the new mails and downloads them. The second step is the decryption process on the user’s Legalmail Client. After the decryption is completed the plain message is shown to the user.

<table>
<tr><th>Use Case Name</th><th>Receive encrypted message</th></tr>
<tr><td><b>ID</b></td><td>E-BUS-LUC-754</td></tr>
<tr><td><b>Main Actor</b></td><td>- Legalmail user</td></tr>
<tr><td><b>Secondary Actors</b></td><td>- LegalmailLegalmail client
- Legalmail service</td></tr>
<tr><td><b>Pre-conditions</b></td><td>- Legalmail user is using a client that support encryption
- Legalmail user have his private key</td></tr>
<tr><td><b>Post-conditions</b></td><td>- The message decrypted and can be read</td></tr>
<tr><td><b>Description</b></td><td>- Legalmail user Bob receives an encrypted message sent by another user</td></tr>
<tr><td><b>Image</b></td><td>


</td></tr>
</table>

C.1.1.5 Receive forwarded re-encrypted message

This use case explains how a forwarded mail is received by the Legalmail Client. First, the Legalmail System checks for any incoming mail and if a forwarding rule exists. For each forwarding rule the mail will be re-encrypted and stored in the recipient's mailbox. After that the new e-mails in the mailbox are fetched by the Legalmail Client.

Use Case Name	Receive forwarded re-encrypted message
ID	E-BUS-LUC-755
Main Actor	- Legalmail user
Secondary Actors	- Legalmail client - Legalmail service
Pre-conditions	- Legalmail user Bob activated forward rule of encrypted messages - Legalmail user Charlie is using a client that support decryption - Legalmail user Alice sent a mail to Bob
Post-conditions	- The message is read in plain text by Charlie
Description	- Legalmail user Charlie receives a re-encrypted message that has been forwarded after Bob's forward rule activation
Image	 «System Actor» Legalmail System «System Actor» Charlie's legalmail client «Human Actor» Charlie re-encrypt message(RE-K(Charlie)) Store re-encrypted message in Charlie's mailbox Read message Fetch message Re-encrypted message Decrypt message Plain message

C.1.2 Import data into Legalmail subscription form. Choose CREDENTIAL for Authentication.

The use case “Import data into Legalmail subscription form. Choose CREDENTIAL for Authentication” is part of the selected use case “InfoCert e-commerce form filling” as described in Section 6.4.2.

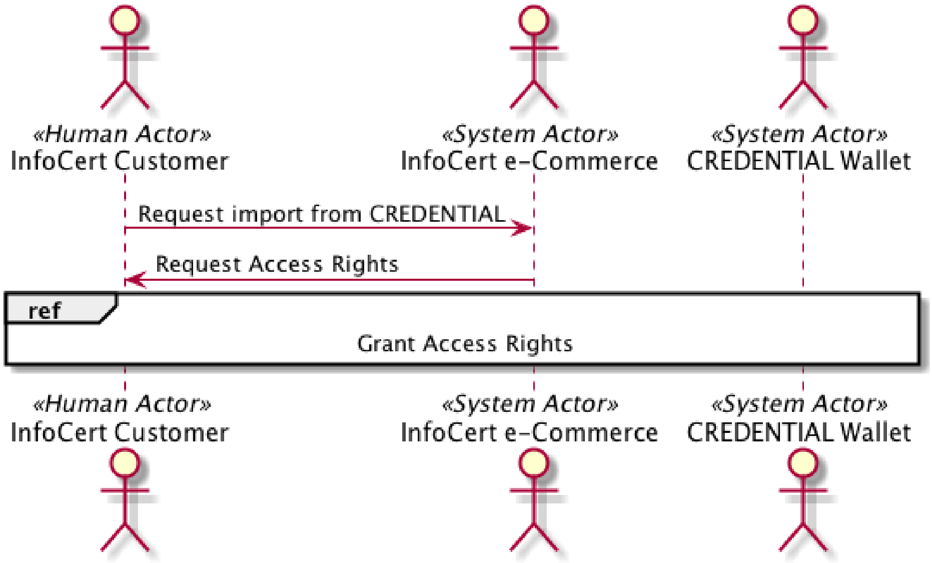
It contains the logical use cases:

- Request data import from wallet
- Import data to InfoCert services form
- Submit filled form

These logical use cases are described in the following subsections.

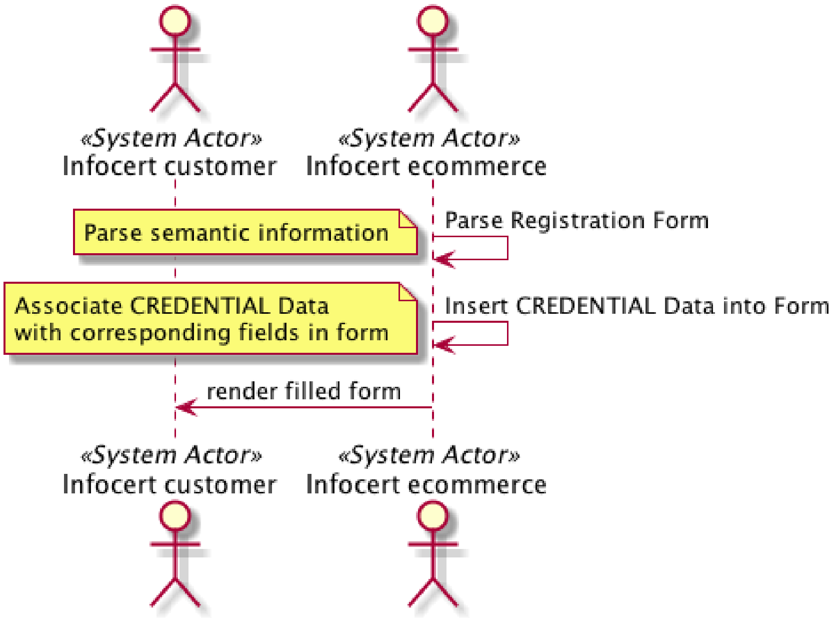
C.1.2.1 Request data import from wallet

The goal of this use case is that an InfoCert user is able to allow access to his CREDENTIAL Wallet data to be used by the InfoCert e-Commerce system. Therefore, access rights for the InfoCert e-Commerce system have to be granted in the CREDENTIAL Wallet. Within this process all cryptographic necessary measurements to enforce data sharing of one participant's CREDENTIAL Wallet data with a third party are taken.

Use Case Name	Request data import from wallet
ID	E-BUS-LUC-757
Main Actor	- InfoCert customer
Secondary Actors	- InfoCert e-commerce - CREDENTIAL Wallet
Pre-conditions	- InfoCert customer has a CREDENTIAL Wallet filled with his data - InfoCert customer accesses InfoCert ecommerce
Post-conditions	- InfoCert e-commerce can import data from wallet
Description	- InfoCert customer wants to fill form with CREDENTIAL Wallet data
Image	 «Human Actor» InfoCert Customer «System Actor» InfoCert e-Commerce «System Actor» CREDENTIAL Wallet Request import from CREDENTIAL Request Access Rights ref Grant Access Rights «Human Actor» InfoCert Customer «System Actor» InfoCert e-Commerce «System Actor» CREDENTIAL Wallet

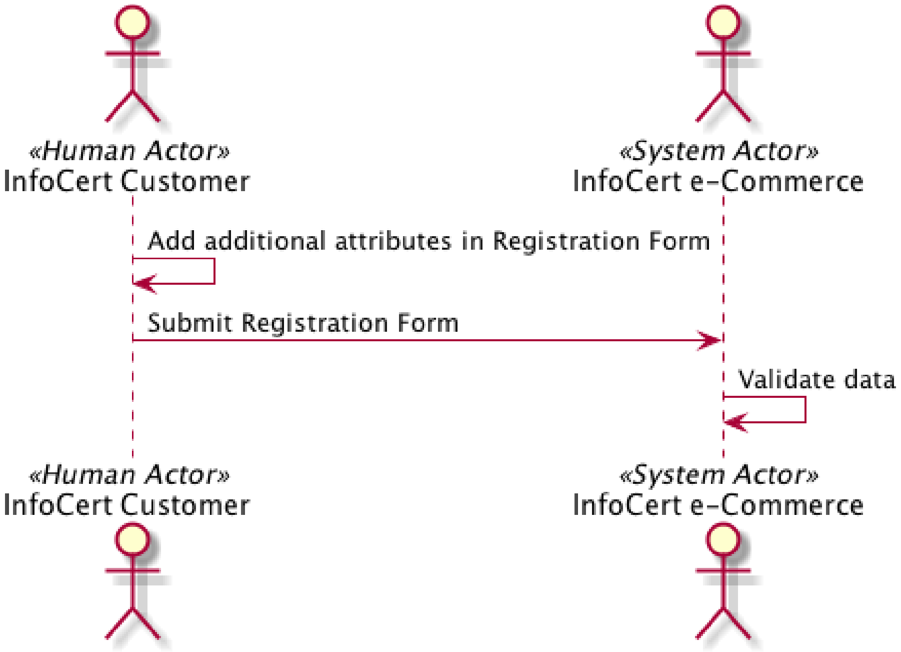
C.1.2.2 Import data to InfoCert services form

This use cases describes how the data received by e-Commerce system are parsed and provided in the registration form for the services registration. The data are then returned to the user that is able to verify and maybe edit such data.

Use Case Name	**Import data for InfoCert services form**
ID	E-BUS-LUC-756
Main Actor	- InfoCert customer
Secondary Actors	- InfoCert e-commerce - CREDENTIAL Wallet
Pre-conditions	- InfoCert customer has access to InfoCert e-commerce - InfoCert customer requests an import from CREDENTIAL
Post-conditions	- InfoCert customer is registered to ecommerce or requests a new LegalmailLegalmail mailbox
Description	- InfoCert customer wants to register to e-commerce or to request a LegalmailLegalmail mailbox
Image	 «System Actor» Infocert customer «System Actor» Infocert ecommerce Parse semantic information Parse Registration Form Associate CREDENTIAL Data with corresponding fields in form Insert CREDENTIAL Data into Form render filled form «System Actor» Infocert customer «System Actor» Infocert ecommerce

C.1.2.3 Submit filled form

This use case describes the final registration step with the CREDENTIAL Wallet provided data. After the user receives the prefilled form, he is able to make additional changes or add more attributes to the registration form. The form is then submitted to the e-Commerce system, validated, and in case of successful validation the user is registered to the system.

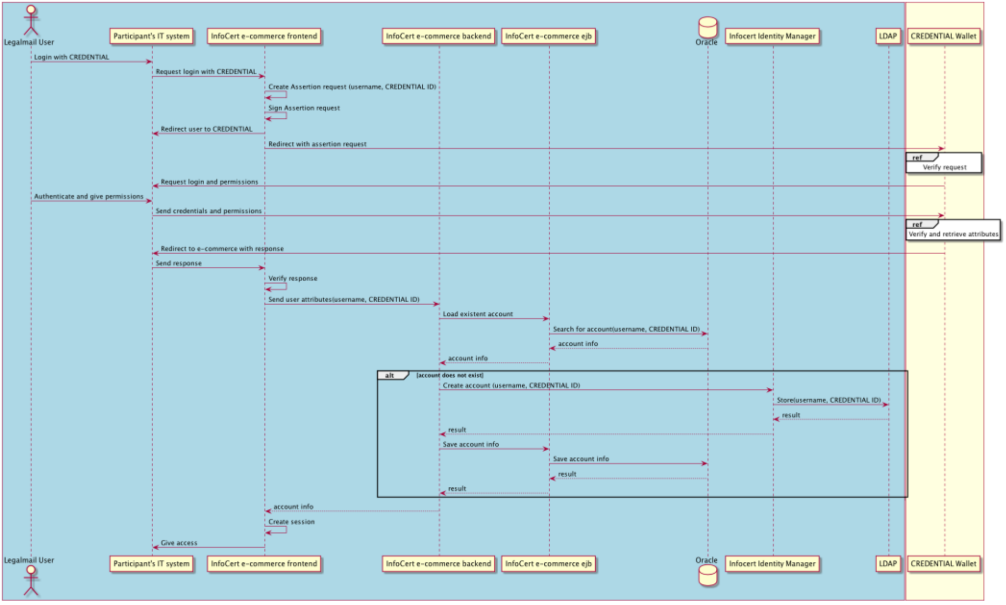
Use Case Name	**Submit filled form**
ID	E-BUS-LUC-758
Main Actor	- InfoCert customer
Secondary Actors	- InfoCert e-commerce
Pre-conditions	- InfoCert Customer has received an ecommerce form filled with data stored in the user's CREDENTIAL Wallet
Post-conditions	- The form has been correctly submitted
Description	- InfoCert customer can complete registration form with additional data and submit to ecommerce
Image	 «Human Actor» InfoCert Customer «System Actor» InfoCert e-Commerce Add additional attributes in Registration Form Submit Registration Form Validate data «Human Actor» InfoCert Customer «System Actor» InfoCert e-Commerce

C.2 Technical Use Cases

The description of the technical use cases for the eBusiness pilot has the goal to describe how the InfoCert internal systems interact with the CREDENTIAL Wallet. Therefore, an understanding of which interfaces need to be integrated as well as which standards and protocols can be used is needed. Thus, the technical use cases refine the logical use cases as described in Section C.1.

C.2.1 Login to InfoCert e-commerce with CREDENTIAL

This use case describes the integration of a CREDENTIAL Wallet Identity Management System within the InfoCert e-commerce services. Thus, the user uses his CREDENTIAL Wallet identity, authenticates himself to the CREDENTIAL Wallet and uses this identity to authenticate to the CREDENTIAL Wallet. This use case shows, that the integration of a CREDENTIAL Identity Provider and a CREDENTIAL Attribute Provider has to be performed in order to authenticate and share identity attributes with an external service.

Use Case Name	**Login to InfoCert e-commerce with CREDENTIAL**
ID	E-BUS-TUC-001
Main Actor	- InfoCert user
Secondary Actors	- InfoCert e-commerce - InfoCert Identity Manager - CREDENTIAL Wallet
Pre-conditions	- The user lands on the InfoCert e-commerce website and sees the CREDENTIAL link for the login with his CREDENTIAL Account
Post-conditions	- The user is logged in the InfoCert e-commerce and, if he didn't already have an account connected to CREDENTIAL, a new e-commerce account is created and linked to CREDENTIAL account
Descriptio n	- The user can choose to login to the InfoCert e-commerce using his CREDENTIAL account, maintaining the same username and password. He follows the CREDENTIAL link and authenticates to his wallet, giving permission for the username and CREDENTIAL account ID requested by InfoCert e-commerce and is redirected back to the e-commerce website. The CREDENTIAL account shall be connected with an InfoCert account, if he didn't have one, the e-commerce will create a new InfoCert account using the username requested to the wallet and will connect it to the account ID provided by the wallet. After the successful login, the user can request services using the InfoCert account.
Image	

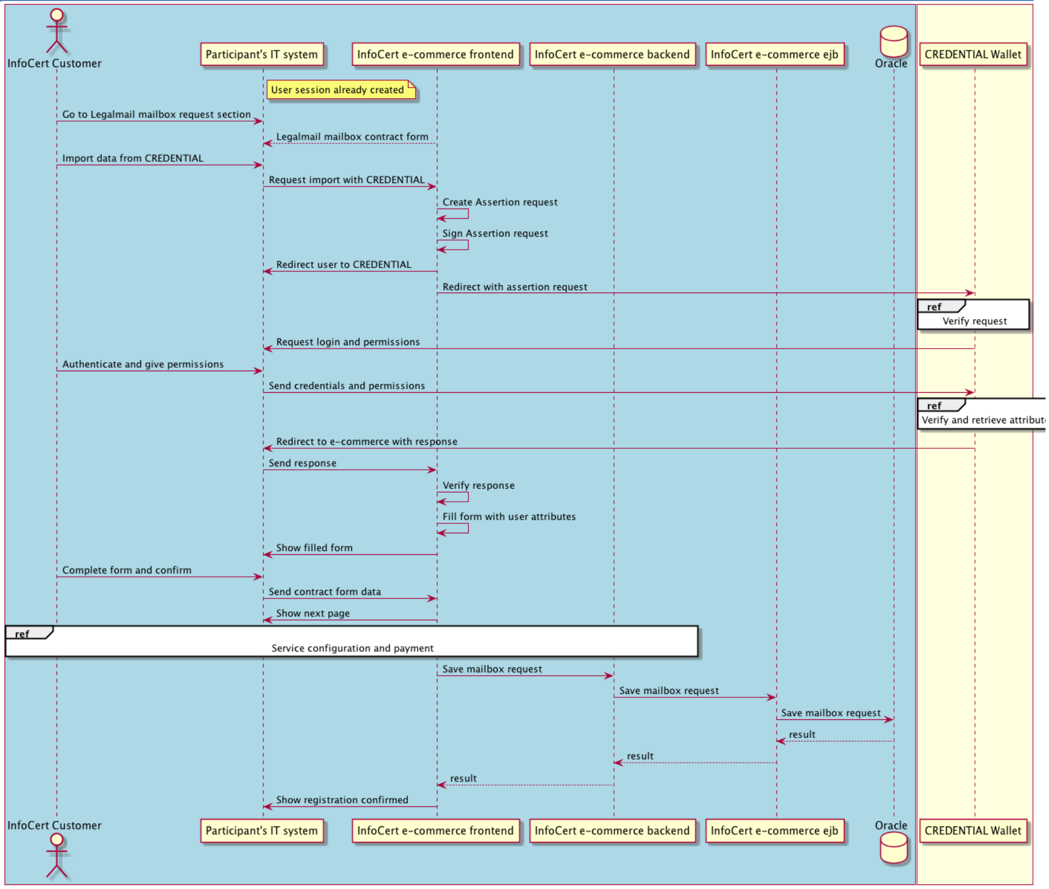
Legalmail User
Participant's IT system
InfoCert e-commerce frontend
InfoCert e-commerce backend
InfoCert e-commerce ejb
Oracle
Infocert Identity Manager
LDAP
CREDENTIAL Wallet
Login with CREDENTIAL
Request login with CREDENTIAL
Create Assertion request (username, CREDENTIAL ID)
Sign Assertion request
Redirect user to CREDENTIAL
Redirect with assertion request
ref
Verify request
Request login and permissions
Authenticate and give permissions
Send credentials and permissions
ref
Verify and retrieve attributes
Redirect to e-commerce with response
Send response
Verify response
Send user attributes(username, CREDENTIAL ID)
Load existent account
Search for account(username, CREDENTIAL ID)
account info
account info
alt
[account does not exist]
Create account (username, CREDENTIAL ID)
Store(username, CREDENTIAL ID)
result
result
Save account info
Save account info
result
result
account info
Create session
Give access

C.2.2 Import data from the wallet to fill Legalmail contract form

This technical use case describes the special interaction with the Legalmail service in order to request a new Legalmail account. The CREDENTIAL Identity Provider and the CREDENTIAL Attribute Provider services need to be integrated in order to fulfill the special needs of the Legalmail registration process.

Use Case Name	**Import data from the wallet to fill Legalmail contract form**
ID	E-BUS-TUC-002
Main Actor	- InfoCert customer
Secondary Actors	- InfoCert e-commerce - CREDENTIAL Wallet
Pre-conditions	- The user is logged to the InfoCert e-commerce and goes to the section where a Legalmail mailbox can be requested. The user goes to the step where the contract form is displayed and sees the CREDENTIAL link
Post-conditions	- The user can submit the form, partially filled with data imported from the wallet
Description	- A user logins to the InfoCert e-commerce and wants to request a new Legalmail mailbox, so he goes to the Legalmail section and chooses to start the process to request a new mailbox. The user follows all the steps required until he needs to fill the Legalmail contract form, where he can choose to import some data from his CREDENTIAL Wallet. The user follows the link to import his data from CREDENTIAL, where he authenticates and gives permissions to import the personal data requested by InfoCert e-commerce. After giving permissions, he is redirected back to the e-commerce website and sees the Legalmail contract form partially filled. The user completes the form with the missing data and proceeds to the next steps, service configuration and payment. In the final step, the user sees the Legalmail mailbox registration confirmation.

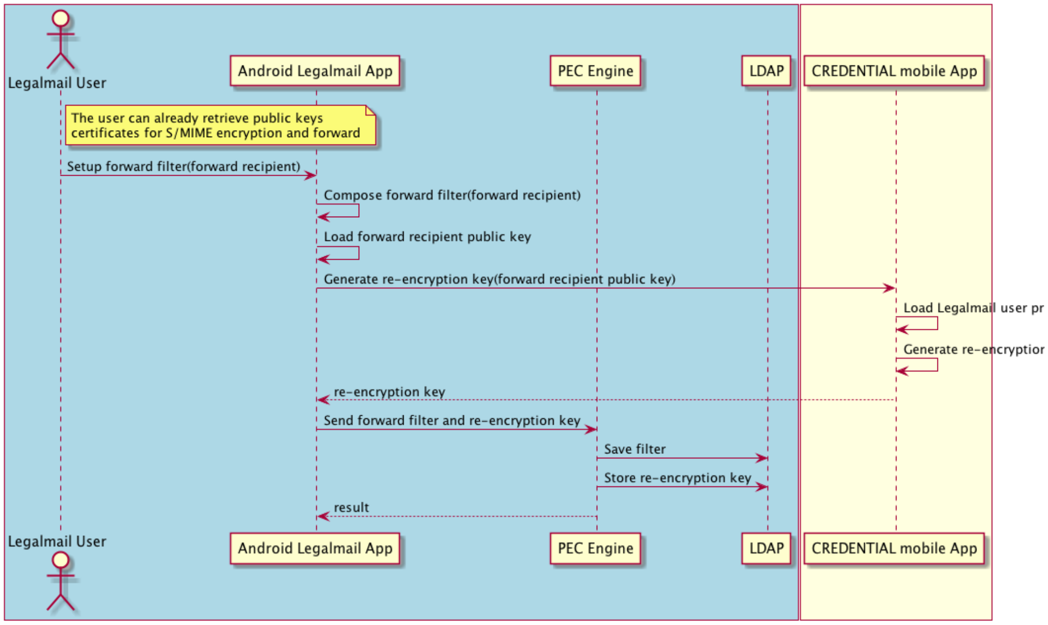
Image


InfoCert Customer
Participant's IT system
InfoCert e-commerce frontend
InfoCert e-commerce backend
InfoCert e-commerce ejb
Oracle
CREDENTIAL Wallet
User session already created
Go to Legalmail mailbox request section
Legalmail mailbox contract form
Import data from CREDENTIAL
Request import with CREDENTIAL
Create Assertion request
Sign Assertion request
Redirect user to CREDENTIAL
Redirect with assertion request
ref
Verify request
Request login and permissions
Authenticate and give permissions
Send credentials and permissions
ref
Verify and retrieve attribut
Redirect to e-commerce with response
Send response
Verify response
Fill form with user attributes
Show filled form
Complete form and confirm
Send contract form data
Show next page
ref
Service configuration and payment
Save mailbox request
Save mailbox request
Save mailbox request
result
result
result
Show registration confirmed
InfoCert Customer
Participant's IT system
InfoCert e-commerce frontend
InfoCert e-commerce backend
InfoCert e-commerce ejb
Oracle
CREDENTIAL Wallet

C.2.3 Legalmail encrypted mail forward filter configuration

This use case describes the technical integration between the Legalmail application for Android devices and the CREDENTIAL Wallet Android App. The CREDENTIAL Wallet Android App is used to create the re-encryption key which is necessary to create the forward rule for the Legalmail scenario. After retrieving the re-encryption key, the Legalmail app sends to the PEC engine server the data composed for the filter and the re-encryption key. The PEC Engine saves both the filter and the associated re-encryption key in an LDAP.

Use Case Name	**Legalmail encrypted mail forward filter configuration**
ID	E-BUS-TUC-003
Main Actor	- Legalmail user
Secondary Actors	- Android Legalmail client - Android CREDENTIAL mobile App - Legalmail PEC Engine
Pre-conditions	- The user has configured his Legalmail account in the Android Legalmail client and wants to setup a mail forward filter to forward encrypted messages to a trusted person. The user has already imported the public key certificate of the trusted person to his client.
Post-conditions	- The mail forward configuration filter for encrypted messages is setup to forward messages to a trusted person.
Description	- The Legalmail user wants to configure a mail forward filter for encrypted message that he receives from a specific sender. The user accesses the filter configuration section of his Android Legalmail app and setups the configuration choosing a trusted user that will be the forward recipient. The Android Legalmail App, first composes the filter, then loads the forward recipient's public key and sends it to the CREDENTIAL mobile app requiring the re-encryption key generation for that recipient. The CREDENTIAL mobile app will load the Legalmail user private key, generate the re-encryption key and return it to the Android Legalmail app. The Legalmail app will send to the PEC engine server the data composed for the filter and the re-encryption key. The PEC Engine will save both the filter and the associated re-encryption key in an LDAP, returning the response of the operation to the Legalmail app.

Image	 -

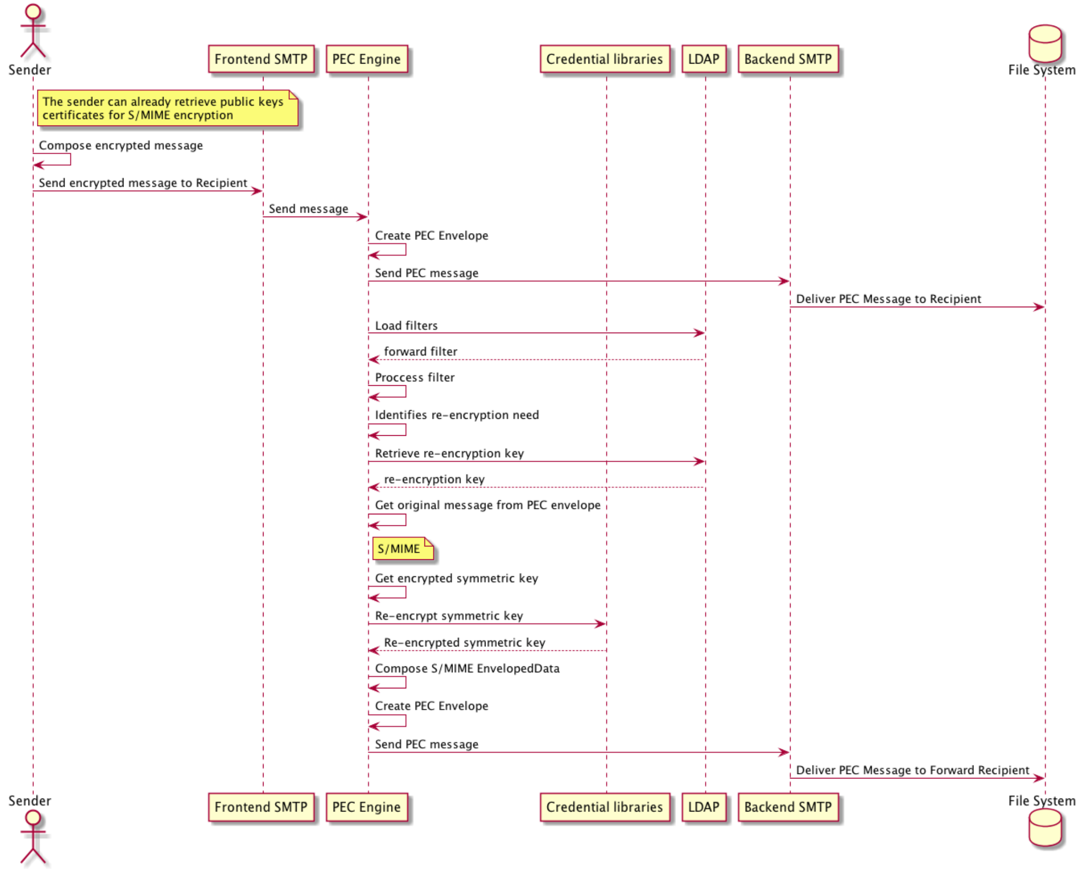
Legalmail User
Android Legalmail App
PEC Engine
LDAP
CREDENTIAL mobile App
The user can already retrieve public keys certificates for S/MIME encryption and forward
Setup forward filter(forward recipient)
Compose forward filter(forward recipient)
Load forward recipient public key
Generate re-encryption key(forward recipient public key)
Load Legalmail user pr
Generate re-encryptior
re-encryption key
Send forward filter and re-encryption key
Save filter
Store re-encryption key
result
Legalmail User
Android Legalmail App
PEC Engine
LDAP
CREDENTIAL mobile App

C.2.4 Legalmail encrypted mail forward

This use case highlights the integration between S/MIME protocol and the decryption of encrypted mails via the CREDENTIAL libraries.

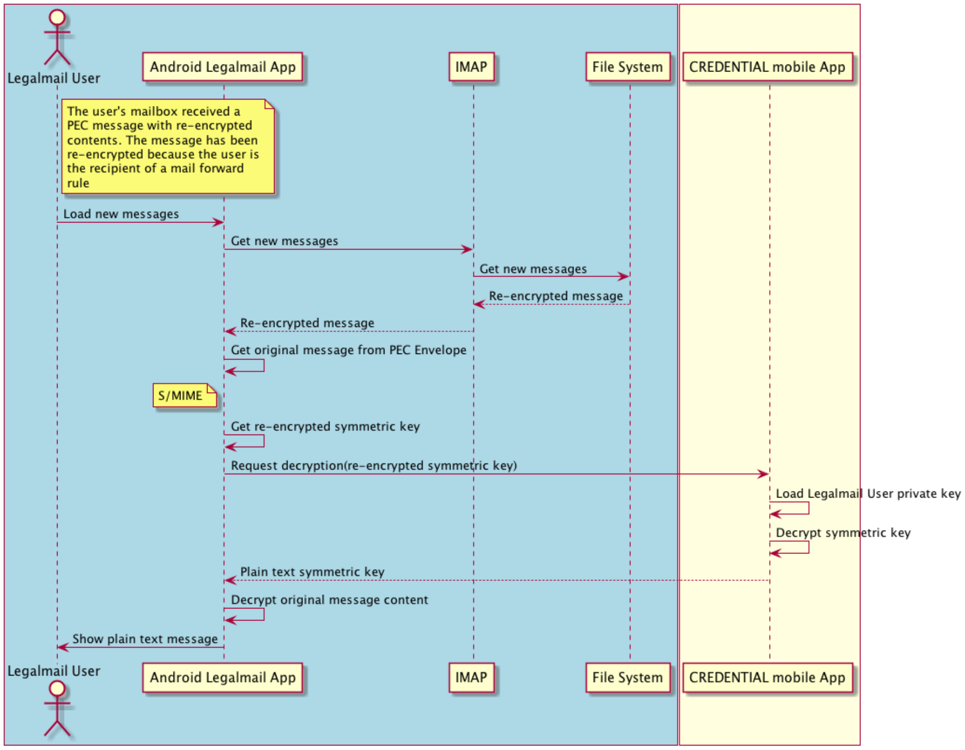
Use Case Name	**Legalmail encrypted mail forward**
ID	E-BUS-TUC-004
Main Actor	- Legalmail user
Secondary Actors	- Legalmail PEC Engine - Legalmail frontend SMTP - Legalmail backend SMTP
Pre-conditions	- The Legalmail user has successfully setup a mail forward filter for encrypted messages received by a specific sender that sent a message
Post-conditions	- The re-encrypted message has been stored in the forward recipient mailbox
Description	- The sender sends an encrypted message to the Legalmail user that wants that message to be forwarded to a trusted user. The Legalmail frontend SMTP receives the message composed by the sender and forwards it to the PEC Engine. The PEC Engine creates the PEC Envelope following the PEC protocol and sends the PEC message to the backend SMTP to be delivered to the Legalmail user. After that, the PEC Engine loads from an LDAP the filters and recognizes that a forward filter for encrypted messages has been configured. The PEC Engine loads the re-encryption key needed from the LDAP and re-encrypts, using the CREDENTIAL libraries, the encrypted symmetric key, that is part of the message as expected by the S/MIME protocol. The PEC Engine composes the parts of the message including the re-encrypted symmetric key and envelopes the original message in a PEC envelope. Finally the PEC Engine forwards the message to the backend SMTP that delivers the message to the forward recipient's mailbox.

Image


Sender
Frontend SMTP
PEC Engine
Credential libraries
LDAP
Backend SMTP
File System
The sender can already retrieve public keys certificates for S/MIME encryption
Compose encrypted message
Send encrypted message to Recipient
Send message
Create PEC Envelope
Send PEC message
Deliver PEC Message to Recipient
Load filters
forward filter
Proccess filter
Identifies re-encryption need
Retrieve re-encryption key
re-encryption key
Get original message from PEC envelope
S/MIME
Get encrypted symmetric key
Re-encrypt symmetric key
Re-encrypted symmetric key
Compose S/MIME EnvelopedData
Create PEC Envelope
Send PEC message
Deliver PEC Message to Forward Recipient
Sender
Frontend SMTP
PEC Engine
Credential libraries
LDAP
Backend SMTP
File System

C.2.5 Visualize a Legalmail encrypted mail forwarded

This use case shows how the Android Legalmail App is integrated with the CREDENTIAL mobile App in order to visualize a received encrypted mail. The received mail content is propagated to the CREDENTIAL mobile App and decrypted. The decrypted content is embedded in the received mail and rendered through the common mechanisms of the Legalmail App.

Use Case Name	Visualize a Legalmail encrypted mail forwarded
ID	E-BUS-TUC-005
Main Actor	- Legalmail user
Secondary Actors	- Android Legalmail App - Android CREDENTIAL mobile App - Legalmail IMAP Server
Pre-conditions	- A re-encrypted PEC message has been delivered by Legalmail system to the forward recipient's mailbox. The Legalmail user wants to read the message using the Android Legalmail client.
Post-conditions	- The Legalmail user can read the plain text message displayed by the Android Legalmail app
Description	- The Legalmail user uses the Android Legalmail App to load new messages and receive a re-encrypted message. The Legalmail app get the original message attached to the PEC envelope and, following the S/MIME protocol, extract the re-encrypted symmetric key used to encrypt the body. The Legalmail app send the re-encrypted key to the CREDENTIAL mobile app which load the user0sprivate key and decrypt the key returning it in plain text. The Legalmail app use the plain text symmetric key to decrypt the body of the message and display it to the user
Image	 Legalmail User Android Legalmail App IMAP File System CREDENTIAL mobile App The user's mailbox received a PEC message with re-encrypted contents. The message has been re-encrypted because the user is the recipient of a mail forward rule Load new messages Get new messages Get new messages Re-encrypted message Re-encrypted message Get original message from PEC Envelope S/MIME Get re-encrypted symmetric key Request decryption(re-encrypted symmetric key) Load Legalmail User private key Decrypt symmetric key Plain text symmetric key Decrypt original message content Show plain text message Legalmail User Android Legalmail App IMAP File System CREDENTIAL mobile App -