Architecture Document

Table of Contents

1. Foreword

This document describes the architecture of the EUU/EAA system.

The document is structured as follows:

No.	Name	Audience	Description
X	Module Name	General interest	A general textual description of the module.
X.1	Thoughts behind	General interest	The ideas behind the module. Should give a quick and comprehensive overview.
X.2	Use Cases	Technical interest	A use case analysis of the module. Consists of an overview and detailed descriptions of each use case.
X.3	Specification	Technical interest	The specification outlines technical constraints of the module and workflows involved.
X.4	Components	Technical interest	Gives an overview of software components to be used and a detailed description of each component.

For the non-IT-specialists a glossary can be found at the end of this document explaining some of the technical terms used in this paper.

2. Introduction

The goal of this specification is the definition of common IT tools for the users of the European neutron and photon research facilities, providing them with novel possibilities for easy access to the facilities.

Currently, there are more than 30’000 users performing experiments at the presently about two dozen existing European photon and neutron facilities. In most cases, these facilities are embedded in national research institutions, performing research in a wide range of disciplines, like materials sciences, life sciences, physics, chemistry, environmental sciences, and studies in cultural heritage.

A normal sequence for an experiment begins with several scientists forming a collaboration and submitting a research proposal (formal document) to one facility or to several facilities. Each facility has proposal review committees, which select in regular intervals the best proposals and assign beamtime to them. If successful, the users then come to the facility (typically for a few days) and perform their experiment with the results published in standard scientific journals. For handling all the administrative issues, each facility runs its own user office manned with several persons and with a Web-based User Office (WUO) as IT backbone. In Europe, these WUOs are based currently on three systems; DUO-type (PSI, HZB, DESY , SOLEIL , FRM II[♣] , MAX-lab[♣]), SMIS-type (ESRF , DIAMOND, ANKA) and VUO-type (Elettra). An essential part of each system is the user database. Currently, there is no connection between the various databases of the different facilities and users have to register individually to each facility of interest with all the related administrative overhead.

Two FP7 roadmap projects (ESRFUP, IRUVX) contain initiatives for a common EU-wide user service. At an early stage, the two projects joined efforts in order to avoid the development of two similar but separate systems and catering to almost the same user community. It has been decided to implement a prototype system as a proof of concept for the specifications contained in this document. Recently, further trans-facility projects (HDRI from the Helmholtz society and PaN-Data and ESFRI Cluster from FP7) are entering the scene. Again, the idea is to join forces and cooperate as much as possible.

The guiding principle of the proposed Umbrella system is that users enter a common web portal before being directed to the appropriate facility. It has to be taken into account, that the facilities concerned are collaborators but at the same time competitors as being, with few exceptions, financed by their national authorities. This collaborator / competitor aspect puts significant constraints on the conceptual design of the common portal.

3. The idea

Irrespective of the implementation details, the basis of the proposed common solution is an EU-wide federated user database. Every user entry will consist of two parts, (a) a central part containing just enough information for being able to define a *user* in a unique way and (b) a local part with further user information, which will remain with local WUOs.

As mentioned before, all European facilities providing beamtime to external users have existing WUOs in operation. All WUO realizations have invested considerable effort to cater for the local environment (e.g. security and accommodation aspects). These investments must be preserved, and a EU-wide authentication approach has to be

[ü] development

[ü]

designed in such a way that it can be interfaced easily to these local WUOs.

As will be shown below, the new system will provide advantages to all the different players:

The *users* will experience novel opportunities for maintaining their user accounts simultaneously at all European facilities. For them, it will be much easier to manage their experiments at these facilities. In addition, the basic preparatory steps are being made for future pan-European tools like remote access to beamlines and data at any of the participating facilities.

The *facility staff* will profit from the fact that users can maintain their information themselves and that in this way the administrative load is reduced and at the same time user information will be more accurate and up to date. In addition, for the first time it will be possible to minimize doublet user registrations –a persistent problem with present WUO systems. Furthermore, a future development of new tools like remote login and data access tools can be made in common, thus optimizing resources.

In respect to the whole *community,* the system will foster the 'community identity'. The system will be able to provide active PR – if the user has registered for that. In addition, it will be easy to extend the system to provide information to the science public and public relation sectors.

3.1. What is the umbrella concept?

The umbrella concept describes a methodology which enhances existing systems with functionality and meanwhile allow full operation of all existing functionality. Following principles are applied:

- Keep existing tools and developments as much as possible
- Put umbrella on top, providing additional functionalities

The following image displays an overview of the whole system:

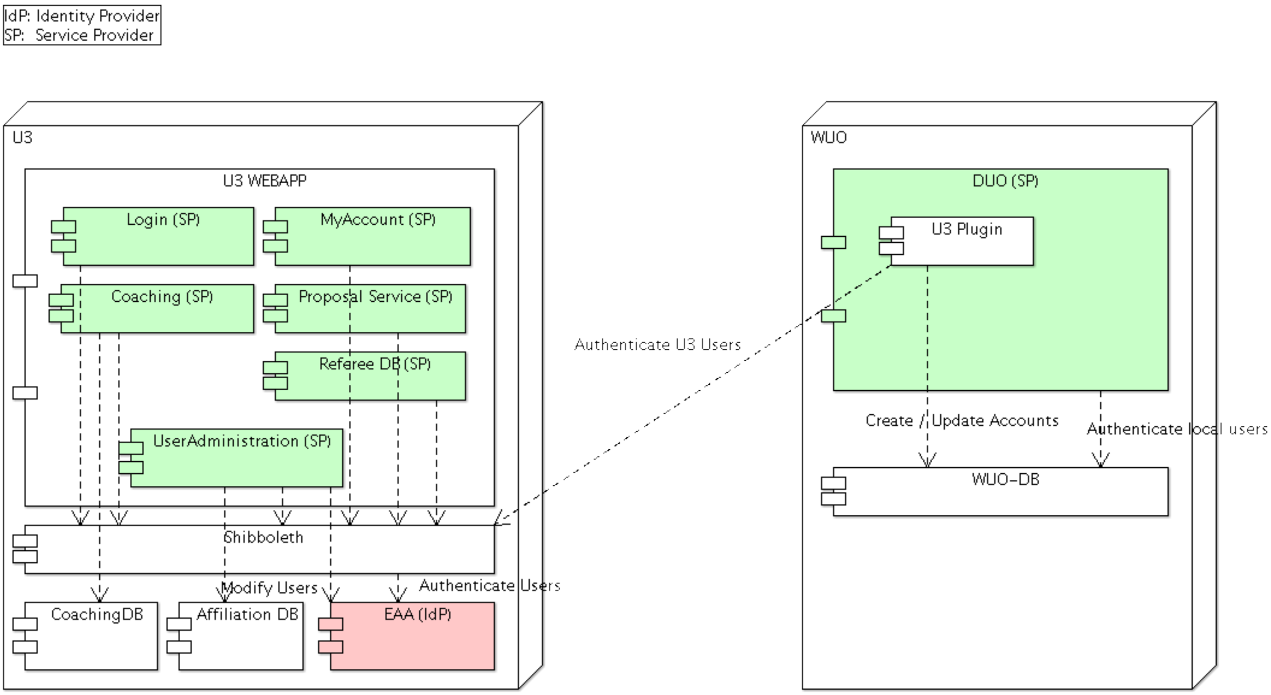


Illustration 1: System overview

4. User Stories

A user story is a software system requirement formulated as sentences in the everyday language of a user. User stories do not act as a contract to the developer but describe situations which should be handled by the software to be implemented.

4.1. Fresh user entering from an university

A freshly graduated student who has never visited a national research facility before wants to perform an experiment at a photon source (e.g. a FEL). As a novel user he/she has no experience how to generate and submit a proposal. Therefore he/she visits the EuroFEL website for further information. He/she finds out that a user account has to be created and starts asking questions first by using the FAQ service and then with the coaching system. After two iterations the proposal is finalized and using the EUU he/she submits the proposal to a proper facility. After being granted beam time at a beamline he/she travels to the facility to pick up the badge prepared by the WUO in advance to his/her visit and to perform his/her experiment.

4.2. Existing facility user moving to another facility

A user group from one facility is going to another facility to perform an experiment. This is increasingly done, because e.g. Users want to study samples with different probes (e.g. photons and neutrons) or special access conditions. Still using the existing credentials the users log in to the new facilities WUO and are transparently authenticated through U3 . In the background a new local user is created with no authorization and matched to the U3 user. The user don't need to request an account at the WUO and after a successful proposal he/she will be authorized by the WUO to access resources.

4.3. Resubmitting a proposal after a period of time

Because of recent technical advances in detectors a team wants to repeat an experiment conducted many years ago at another facility. Unfortunately they lost the old submission but remember at which facility the experiment had taken place. So they log in to the WUO and export the proposal. At U3 they import the proposal, change some parameters and are ready to resubmit the proposal with minimum effort.

4.4. Do I remember my password?

By experience, after some time users tend to forget usernames. When a user decides to submit a proposal 1 hour before the deadline, he/she fortunately needs to remember only one password to access all facilities which may help to submit the proposal in time.

5. EAA, European Authentication and Authorization

A prerequisite for the services of EUU is an EU-wide authentication and authorization concept. The context, in which the system will be used, implies several boundary conditions:

Multiple Trust Levels

There will be a broad spectrum of services. For example, a user may want to register for a workshop or a users meeting. For that, no special security control is required and the administrative threshold should be as low as possible. A soft online registration with an email handshake is sufficient. On the other hand, if a user is requesting a badge at a facility, which will allow him/her to access a restricted site, the management has to be sure about the identity of a user. This can be solved in such a way, that a user with a valid soft registration shows an official identification document at the facility user office and the officer marks the successful user in the database (hard registration). There are different security requirements at the local facilities for different services and the EUU has to provide the flexibility to take that into account. For some facilities it will be sufficient to require a user to perform a once-only hard registration, while other facilities may require for specific services a hard registration with each visit.

User Friendliness

The user friendliness of the system will determine its success. E.g., mobility is high among the users and when they move from one affiliation to another one, it will be highly appreciated if they have to enter their new coordinate only once. To foster both user and facility friendliness, a web based system is a must, in order to circumvent installation of software on local computers.

Increasingly, users are coming to the photon and neutron facilities also from non-traditional research fields like e.g. archaeology or cultural heritage and / or are performing photon / neutron experiments only for a part of their research time. On one side, this is very positive, on the other side, the IT tools provided have to be designed in such a way, that they can be efficiently used also by non-full-time IT experts.

Facility Friendliness

Operational resources for running user offices at the individual facilities are very much limited and it will be important to delegate account maintenance obligations as much as possible to the users. An issue will be the ability of the system to cope with double registrations of users – at least in part manually. This will be mandatory for a unique user identification. The required resources for the modification of the local WUOs will depend clearly on the details of the realization but in any case is it clear that it will not be totally for free. It will probably be of the order of weeks. Technically, this will mean, that the present single identification will have to be modified to a chained identification. Concerning operation, there are obviously two requirements: (a) the new system must be so simple that its use is straightforward also for the occasional user and (b) the operators of the local WUOs will need training.

Compatibility with Existing and Future WUO Authentication Tools

For several reasons, the EAA will have to coexist on a midterm timescale with the authentication mechanisms of the existing WUO tools. In addition, new WUO systems are under development. Therefore, clear interface definitions have to be developed for the communication between the EAA and these tools.

Central and Local User Databases

At present, due to confidentiality issues, a consensus cannot be reached among the participating facilities to centralize all user information. In addition, the new system will have to coexist with the current systems at the local facilities for a long time. The approach, therefore, consist of splitting the user information into a central part, which contains the information necessary for a unique authentication of a user (non-strategic information), and a local part at the respective facility with the rest of the information and the authorization information.

Central Affiliation Database

With very few exceptions, users of the facilities are linked to affiliations. Usually, the affiliation names are complex names without generally accepted definition rules. The only solution is to try a de-facto standard by an agreement between the participating facilities and to define the rules (e.g. language, definition depth) within EAA. The user browses within the list of defined affiliations. If the respective entry is not found the user sends a request to the database operator, who then enters the new item. In addition to the real entry definitions, alias definitions can be added.

5.1. Thoughts behind

- The EAA provides a central login for all participating WUOs.
- To avoid duplicate user entries users must be identified in a unique way. but there is no mathematical guarantee for a duplicate free system.
- The goal is not to replace existing WUO tools but to enrich them with functionality.
- Dual operation of the EAA and the existing WUO-solutions need still to be possible.
- WUOs should stay fully autonomous.
- User information will be kept as much as possible within the present WUO systems, i.e. with the local facilities.
- Only that part will be centralized which is necessary for unique user identification.
- It will not be possible for one facility to access information on a user registered at another facility.
- Storage of personal information is under the control of the user him/herself.
- There will be no central storage of authorization information about a user.
- Facility-related information will remain fully under the control of the respective facility.
- Any speciality of a WUO software should still be able to work.
- There is no need for installation of specific software on user computers.

5.2. Use Cases

In the following a set of functions is defined which the EAA system has to provide. The use cases describe only the expected functionalities but they do not reveal any technical constraints. The figure also shows the actors (roles) associated to the functionality. The Identity Provider and the Service Provider are plain technical actors with no human interaction involved.

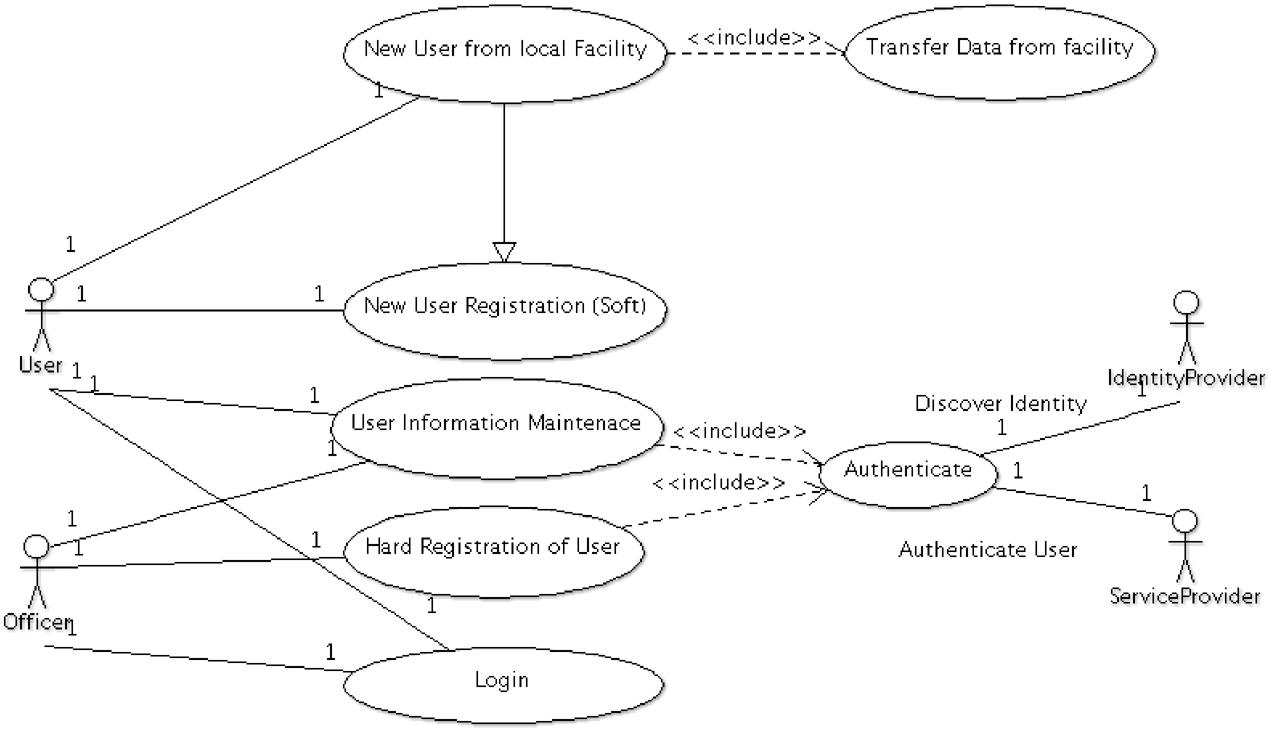

New User from local Facility
<<include>>
Transfer Data from facility
New User Registration (Soft)
User
User Information Maintenace
<<include>>
<<include>>
Authenticate
Hard Registration of User
Login
Officer
Discover Identity
IdentityProvider
Authenticate User
ServiceProvider
1

Illustration 2: Use Cases EAA

Actors	Coaching
User	U3 User
Officer	Responsible for all user-maintenance-related activities
Identity Provider	Directory with user credentials to authenticate against
Service Provider	An application using EAA for authentication

Table 1: Actors EAA

5.2.1. Login

Use Case	UC_EAA1
Name	Login
Identifier	UC_EAA1
Version	0.1
Goal	User is logged in to EAA
Summary	The user is presented with a login mask and after successfully entering the credentials is authenticated against the EAA system.
Actors	User, Identity Provider
Stakeholders	User
Preconditions	User must have an account in EAA to be able to login.
Triggers	• User clicks on login at the EAA website • User wants to use an EUU service
Course of events	1. User is presented with a login screen 2. User enters his credentials 3. Credentials are verified against an identity provider 4. If successful, user is logged in to the EAA system

Table 2: Use Case UC_EAA1

5.2.2. User information maintenance

Use Case	UC_EAA2
Name	User Information Maintenance
Identifier	UC_EAA2
Version	0.2
Goal	Manipulate user information to reflect changes
Summary	In the case of an address mutation a user and/or officer can change his information and disseminate the change to all facilities where he is registered. Nobody has access to all account information. All changes are monitored.
Actors	User, Officer
Stakeholders	User, Officer, Community
Preconditions	User must be registered and logged in at the U3 (Umbrella)
Triggers	• User clicks on “MyAccount” on portal
Course of events	1. User is presented with a form containing his information 2. User manipulates the values to reflect the changes 3. User clicks on “Save” 4. Information request is disseminated to all Service Providers 5. Service provider must return a valid response to show that the user is registered there 6. Information is sent to specific WUO

Table 3: Use Case UC_EAA2

5.2.3. New user registration(soft)

Use Case	UC_EAA3
Name	New user registration(soft)
Identifier	UC_EAA3
Version	0.1
Goal	Register a user with an email handshake
Summary	User registers himself at the EAA and after entering his information receives an email with a link on it to click and verify his email address.
Actors	User
Stakeholders	User
Preconditions	none
Triggers	User clicks on “Create Account” on portal
Course of events	1. User clicks on “Create Account” on portal 2. User receives a form to enter his information 3. User enters his information 4. User clicks on “Save” 5. Data is compared against duplicates. 6. Data is persisted in the directory but marked inactive 7. User receives an email with a link in it. 8. User clicks on link 9. Account in the directory is marked as active

Table 4: Use Case UC_EAA3

5.2.4. New user from local facility

Use Case	UC_EAA4
Name	New user from local facility
Identifier	UC_EAA4
Version	0.1
Goal	A user already existing at a local facility wants to register at the EAA
Summary	A user with an account a local WUO wants to participate at the EAA and have his/her local account linked to his central EAA account
Actors	User
Stakeholders	User
Preconditions	User must be registered at a local facility
Triggers	User clicks on "Register at EAA" on local WUO
Course of events	1. User is logged in at a local WUO 2. User clicks on "Register at EAA" 3. UC_EAA7 is used to transfer information from WUO to EAA 4. UC_EAA3 is used to handle a "soft registration" 5. After successful login to EAA the user is forwarded back to his local WUO. 6. Local WUO reads local session information and the U3 session information to match the U3 account with the local account.

Table 5: Use Case UC_EAA4

5.2.5. Hard registration of user

Use Case	UC_EAA5
Name	Hard registration of user
Identifier	UC_EAA5
Version	0.1
Goal	Hard-register a user after face-to-face verification
Summary	After showing an ID card the hard registration flag is set on the user account after verification in person
Actors	User, Officer
Stakeholders	User, Community
Preconditions	• User must be soft registered at the U3 (Umbrella) • User must show up physically at a facility
Triggers	User shows up with an ID card at a facility
Course of events	1. Officer verifies identity in person 2. Officer sets “hard registration” flag

Table 6: Use Case UC_EAA5

5.2.6. Authenticate

Use Case	UC_EAA6
Name	Authenticate
Identifier	UC_EAA6
Version	0.1
Goal	Authenticate a user against EAA
Summary	In order to use a service provider the user has to be authenticated against the EAA system.
Actors	User
Stakeholders	User, Identity Provider, Service Provider
Preconditions	User must have an account in EAA.
Triggers	User wants to consume a service provider
Course of events	1. User visits a service provider 2. Service provider checks session with the identity provider. 3. If user is not authenticated UC_EAA1 is used to handle login 4. Service provider sets up local session with the associated user

Table 7: Use Case UC_EAA6

5.2.7. Transfer data from local facility

Use Case	**UC_EAA7**
Name	Transfer data from a local facility
Identifier	UC_EAA7
Version	0.1
Goal	Transfer existing user information to EAA
Summary	In order to be user friendly the user can transfer parts of his information from the local facility to the EAA system so that he/she doesn't need to enter the information again when he/she accesses another WUO.
Actors	User
Stakeholders	User
Preconditions	User must be registered at a local facility
Triggers	User clicks on “Register at EAA” on local WUO
Course of events	1. User clicks on “Register at EAA” on local WUO 2. A window opens and shows all available information 3. User selects the attributes to import 4. User is forwarded to EAA registration 5. Form in EAA registration is already filled out

Table 8: Use Case UC_EAA7

5.3. Specification

5.3.1. General

The EAA system is a complex and distributed infrastructure for handling authentication at a super facility level.

Accounts

The idea is to split the user information in two parts: a central part at the EAA containing authentication-relevant information and a local part at the facilities containing in-depth information about roles and access levels at the specific facility (See Illustration: Local and Central Part of User). The local parts are matched to the central part with a hash (See Illustration: Local and Central Part of User as seen from a Database). This requires a directory to store account information on as a central part. Since this list of users is highly sensible, there should be no way to retrieve all of the accounts at once.

Technically, it would be possible to store information like user addresses and email centrally at the EAA, but there are political constraints concerning data security, which have to be taken into account. As a minimal approach, for a unique user identification, name, email and birthdate are required. Therefore it is currently decided, not to save any additional information at the EUU, but to keep the option for future releases, to add additional information in consensus with the community for further services.

Each access to the accounts must be logged and proactively monitored to prevent misuse and fraud. This should happen at the directory level, where users are stored, and a trigger could be a threshold which consists of queries by user by time or a check if someone accesses account information other than his/her own.

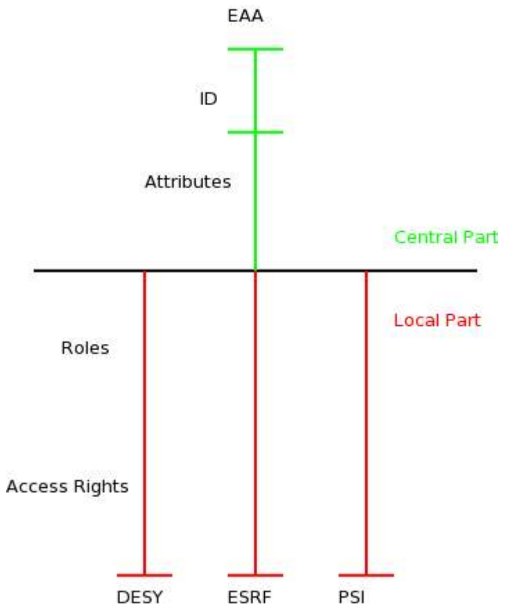


Illustration 3: Local and Central Part of User

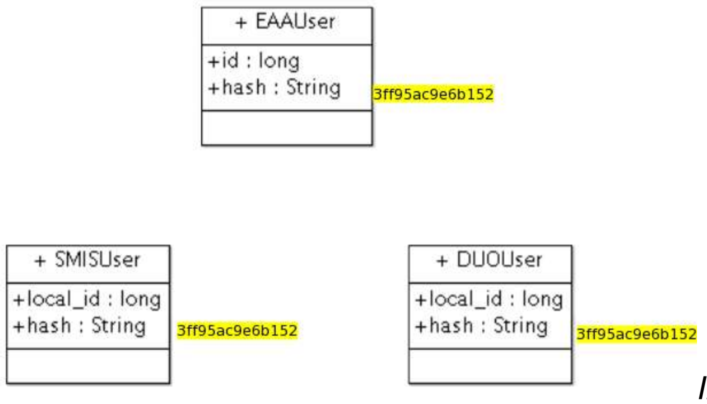


Illustration 4: Local and Central Part of User as seen

Single Sign-On Landscape

At the moment each facility operates its own user database. Users can exist in different facilities and have different user names for each facility (See Illustration: Status Quo).

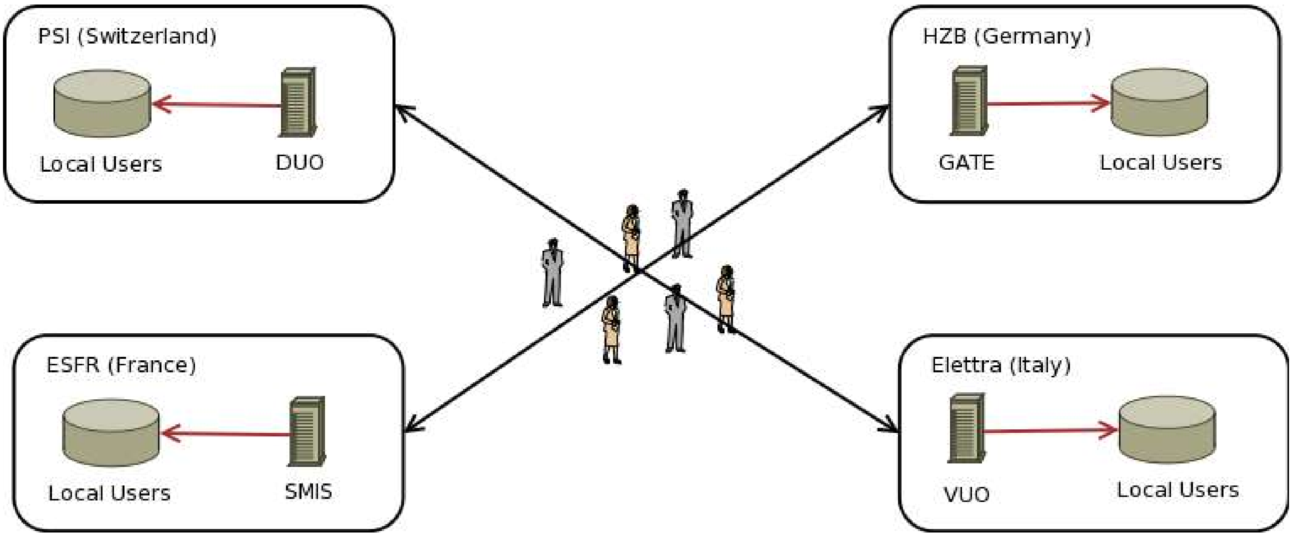


Illustration 5: Status Quo

The EAA system will allow a dual operation either with EAA as an Identity Provider and the WUOs as Service Providers or the WUOs can just use their local user database (Status Quo) to authenticate users.

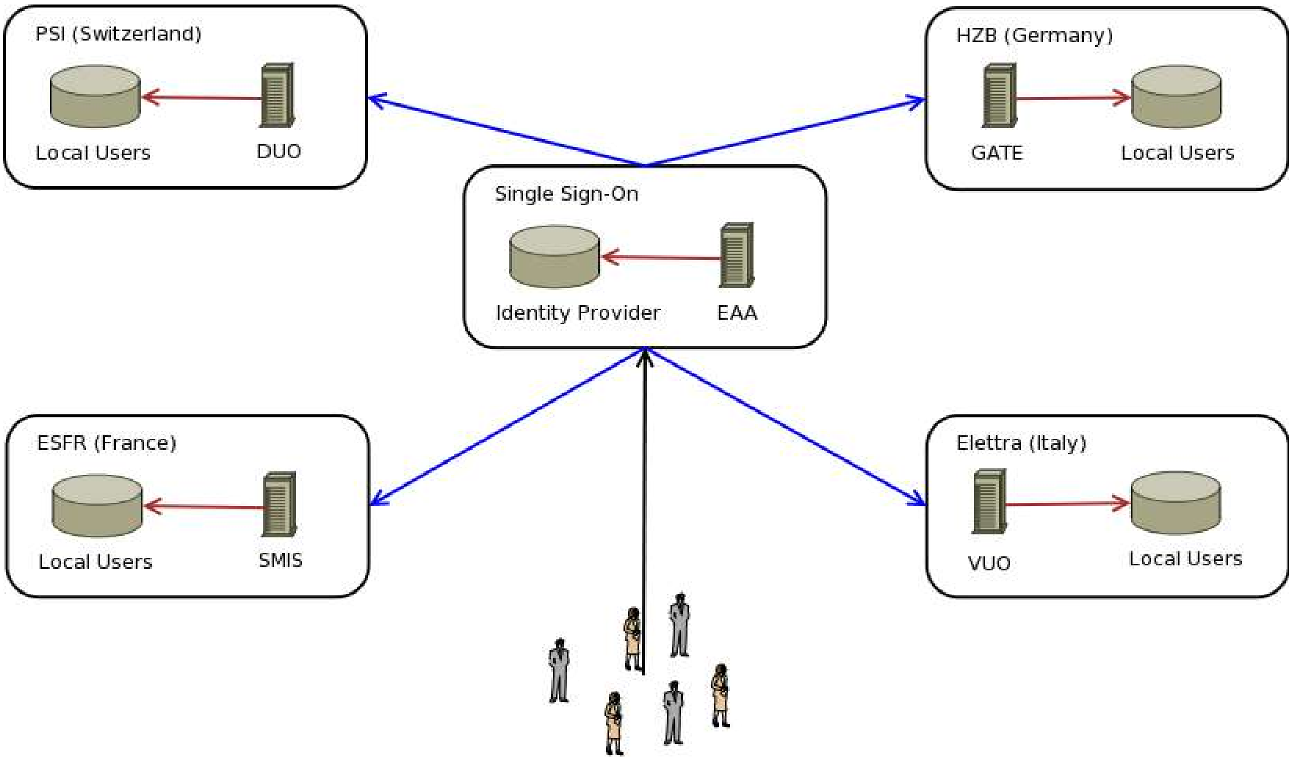


Illustration 6: Landscape with SSO and Dual Operation

5.3.2. Sequence

Status Quo

As seen in the Illustration: Status Quo as a Sequence Diagram the user has to register at every facility and receives a different username at each facility. Address changes must be disseminated to each facility manually.

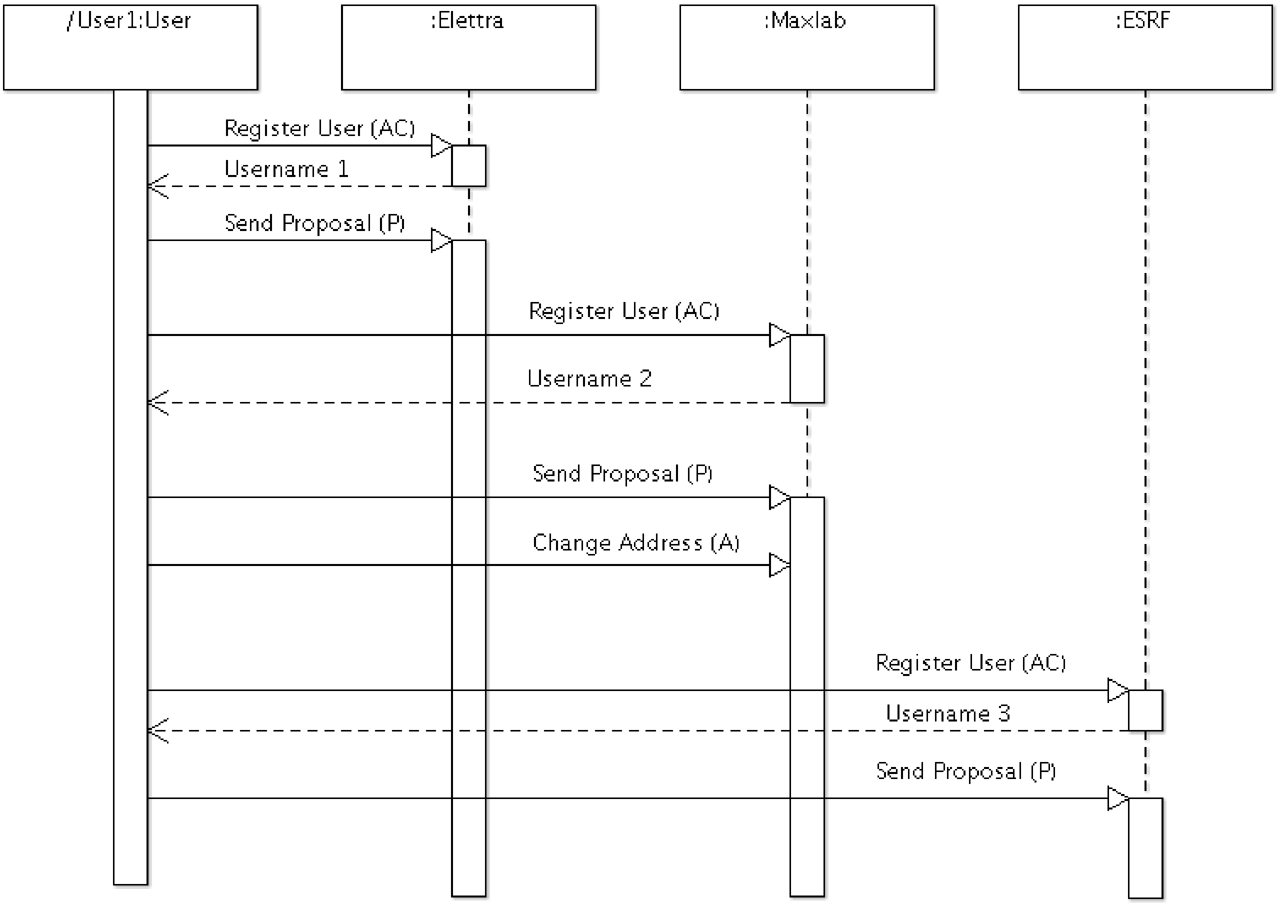
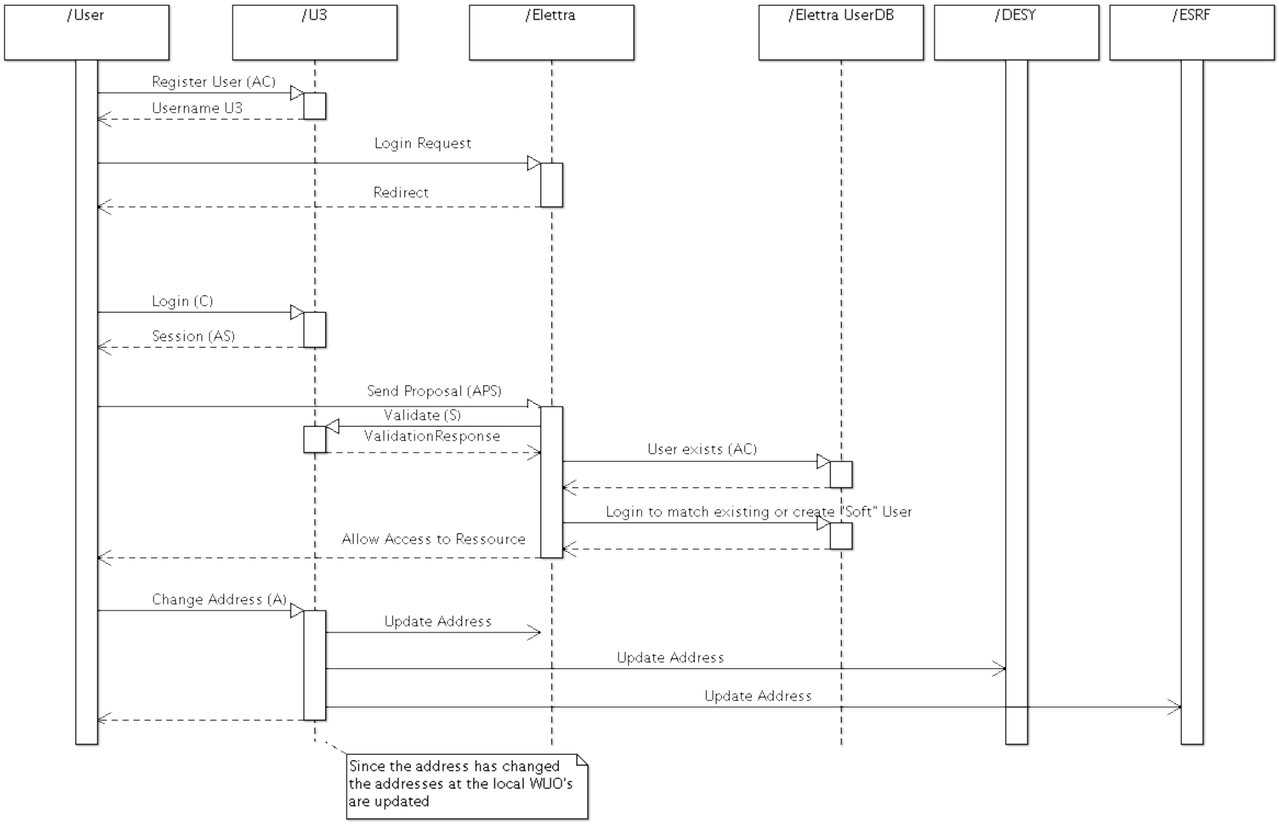


Illustration 7: Status Quo as a Sequence Diagram

EAA Operations

Operations at EAA include only a one-time registration centrally and the creation of local accounts happens transparently when the user connects to a WUO. Address changes are also accomplished transparently on each new connection to a facility. This setup still allows WUOs to create local users with no connection to EAA.



Since the address has changed
the addresses at the local WUO's
are updated

Illustration 8: EAA (Umbrella) System as Sequence Diagram

User Registration

The user request an account at EAA and receives a form for entering the required information. After successful submission of the form the user receives an email with a link on which he/she has to click in order to activate the account. Now the login to the EAA is possible. With registering at EAA users are not automatically created at each WUO – they have to visit the WUO for creating an account.

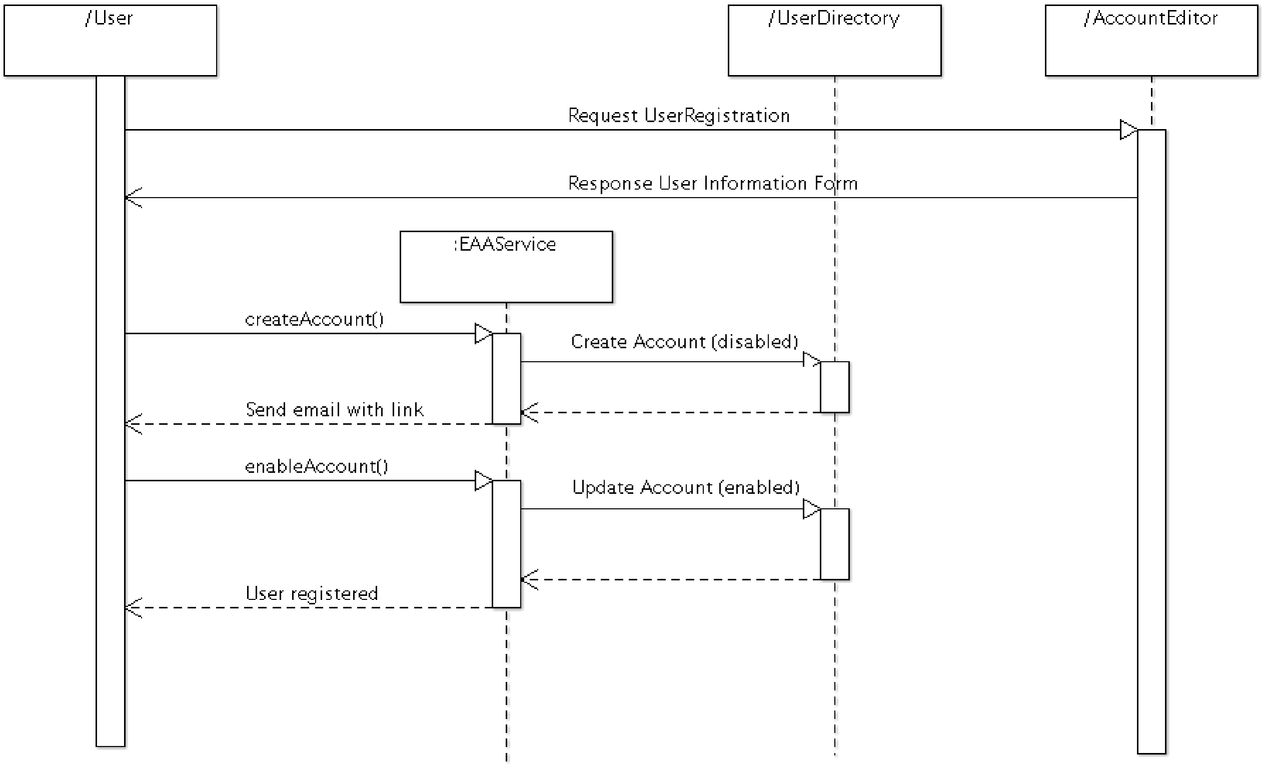


Illustration 9: User Registration as Sequence Diagram

Login – User Creation and Matching

There are many different possible login scenarios during the time of transition to EAA:

- New EAA users with no facility account would like to login to a facility.
- Existing users with facility accounts would like to register at EAA and match their local account to the central account.
- Users with existing EAA account and local accounts would like to match their facility accounts to the EAA account.

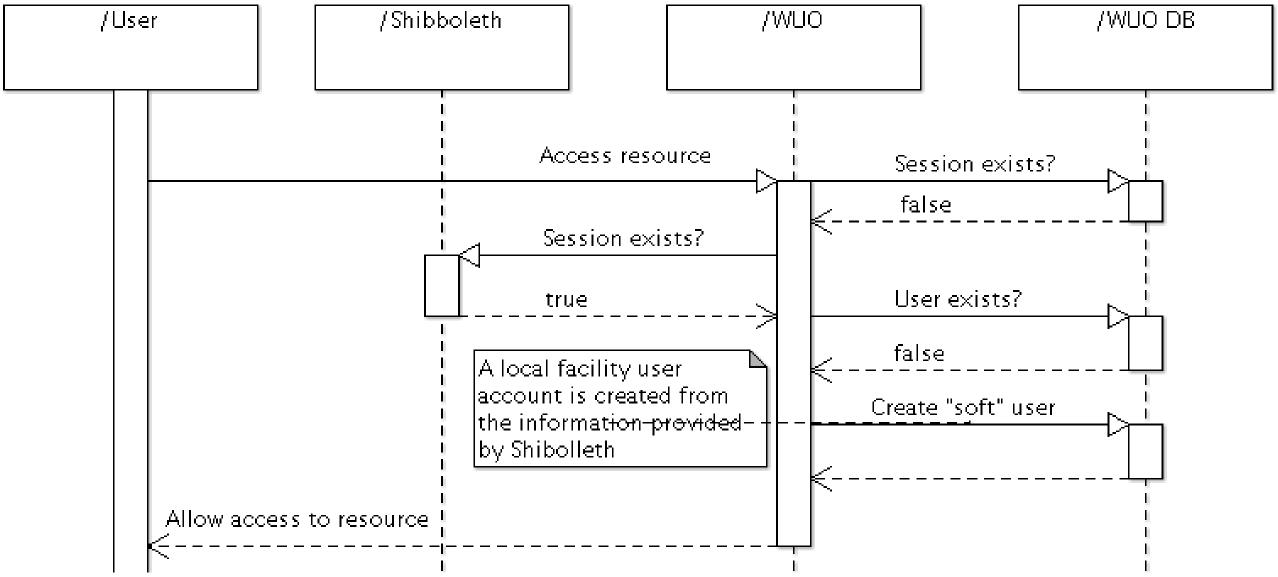


Illustration 10: Login as Sequence Diagram (User exists at EAA but not at WUO)

The Illustration: Login as Sequence Diagram (User exists at EAA but not at WUO) shows the case when a user has an EAA account but no account at a local facility. In this case the user himself is already logged-in at the EAA and tries to access a WUO. After checking the session with EAA the WUO attempts to create a “soft” user at the local WUO DB as the check for a local user failed. The user is created with the attributes passed by EAA in the session and, depending on the local facility policy, is able to use local resources (e.g. submit a proposal)

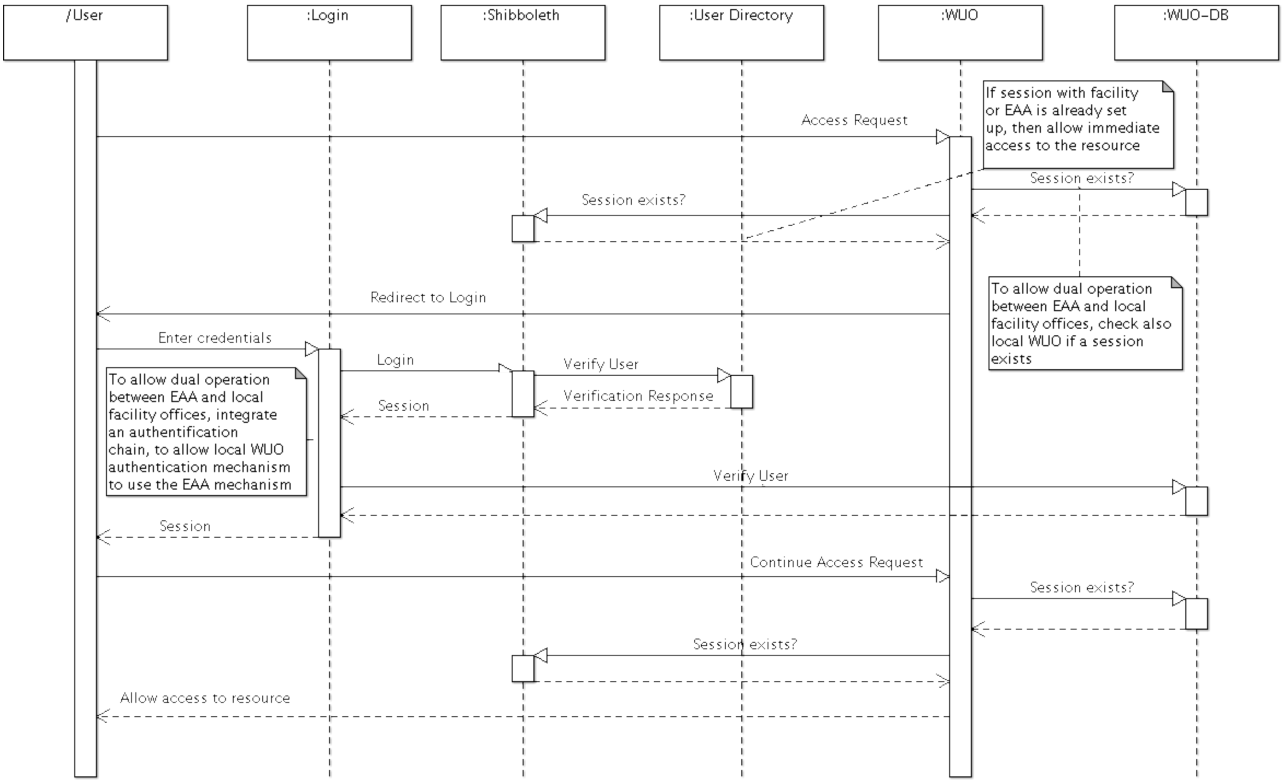

/User
:Login
:Shibboleth
:User Directory
:WUO
:WUO-DB
If session with facility or EAA is already set up, then allow immediate access to the resource
Access Request
Session exists?
Session exists?
To allow dual operation between EAA and local facility offices, check also local WUO if a session exists
Redirect to Login
Enter credentials
Login
Verify User
To allow dual operation between EAA and local facility offices, integrate an authentification chain, to allow local WUO authentication mechanism to use the EAA mechanism
Session
Verification Response
Verify User
Session
Continue Access Request
Session exists?
Session exists?
Allow access to resource

Illustration 11: Login as Sequence Diagram (User exists at WUO and EAA and are matched)

The Illustration: Login as Sequence Diagram (User exists at WUO and EAA and are matched) displays following situation:

- User exists at WUO
- User exists at EAA
- Accounts are matched
- User is not logged in.

As the user attempts to access a resource, the resource itself tries to find an existing session in both WUO and EAA. As the user is not logged-in this fails and the WUO redirects the user to a login. After entering his credentials the system tries to authenticate the user at both EAA and WUO level. On success the user then is forwarded back to the original resource which now has no problems in validating the session and access to the resource is granted.

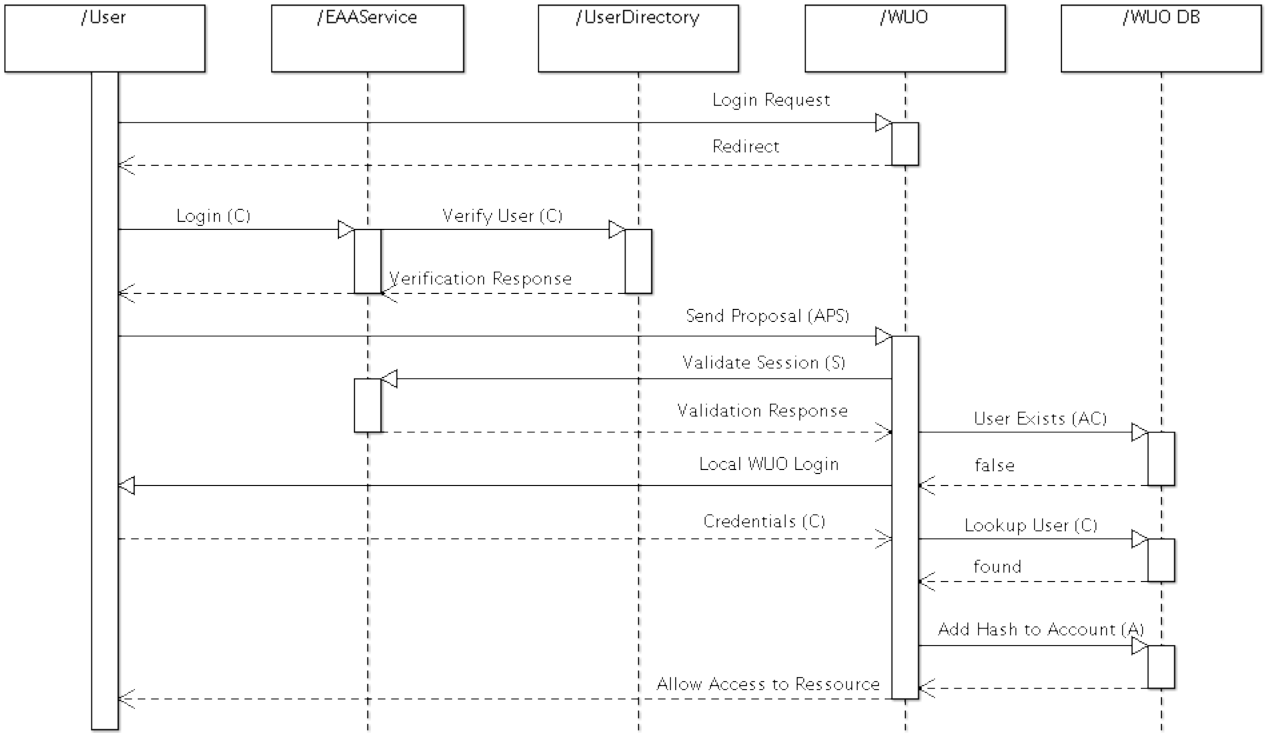


Illustration 12: Login as Sequence Diagram (User exists at WUO and EAA but are not matched)

The Illustration: Login as Sequence Diagram (User exists at WUO and EAA but are not matched) shows following situation:

- User exists at WUO
- User exists at EAA
- Accounts are not matched
- User is not logged in.

The user tries to access a resource but has no active session. He/she then is forwarded to a login where the user enters his/her EAA credentials. After he/she was forwarded back to the resource he/she can decide to either create a local account or (in this case) try to match both accounts. He/she will be presented with a login form to enter the facility credentials and after successfully authenticating against the WUO his EAA hash is inserted into the account which the login retrieved. Beside the hash also the address information passed is used to update the local account. Now the user can access the resource.

Address Mutations

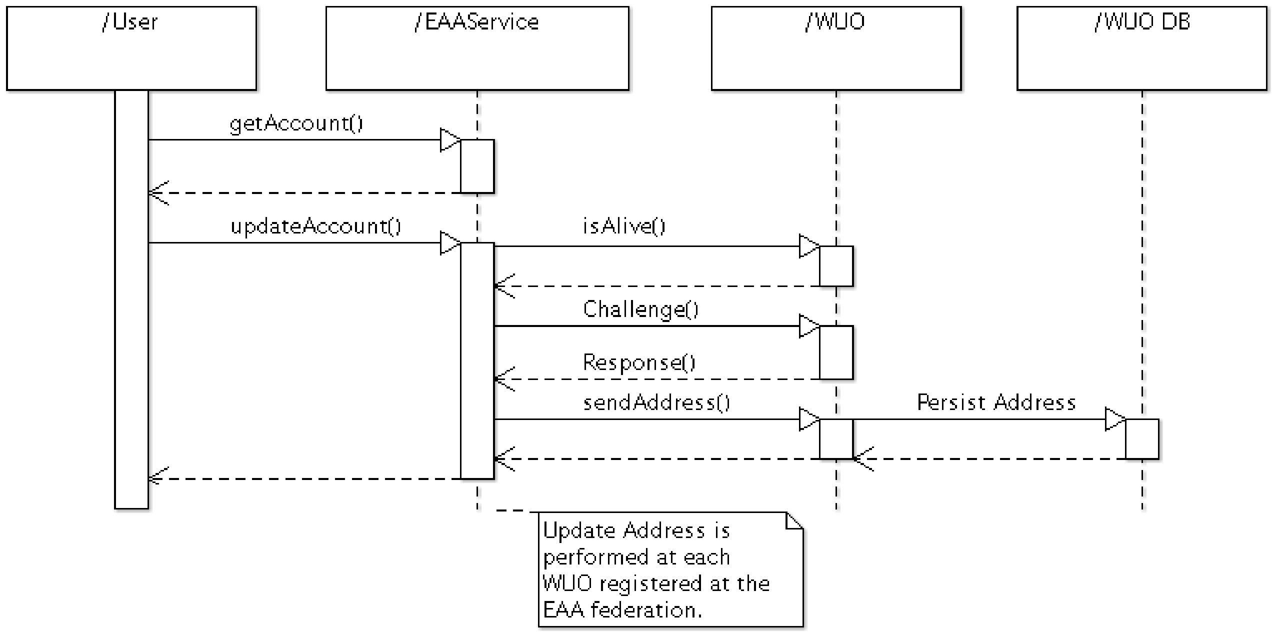


Illustration 13: Address Mutation as Sequence Diagram

Changes of addresses happens centrally at EAA and only one request is needed by the user for initiating the updates at all participating WUOs. For optimum data security, matching is done via a hash associated to the EAA-user. The EAA checks all Service Providers sequentially if they are alive and if not, queues the request for a predefined interval and sends the request again. This measure is taken to prevent information loss in case of a Service Provider downtime. If the Service Provider is alive, a challenge-response authentication is initiated between EAA and WUO to guarantee that the user requesting this update is registered and matched with the WUO.

Challenge-response Authentication for Address Mutations

In order to ensure that a users exists at a WUO and at the same time to constrain the information distribution, cryptographic measures will be used to control the information flow. A symmetric challenge-response authentication concept will be applied.

Besides the Shibboleth hash, the basis for this mechanism is a key associated with the user. The key must always remain secret and protected. It is transported to the WUO by the user as he matches his EAA-account with his WUO-account.

To ensure that a user exist at a WUO, the users Shibboleth hash is sent together with a challenge to the WUO. The WUO retrieves the specific user key from its local database and generates a random number, which is sent back to the EAA. Now both EAA and WUO apply a function to the key, the challenge and the random number and the WUO sends the answer back to EAA. Now EAA verifies the equality of both answers and then sends the updated information to the WUO. An answer must be returned in a timeout period in order to prevent attacks.

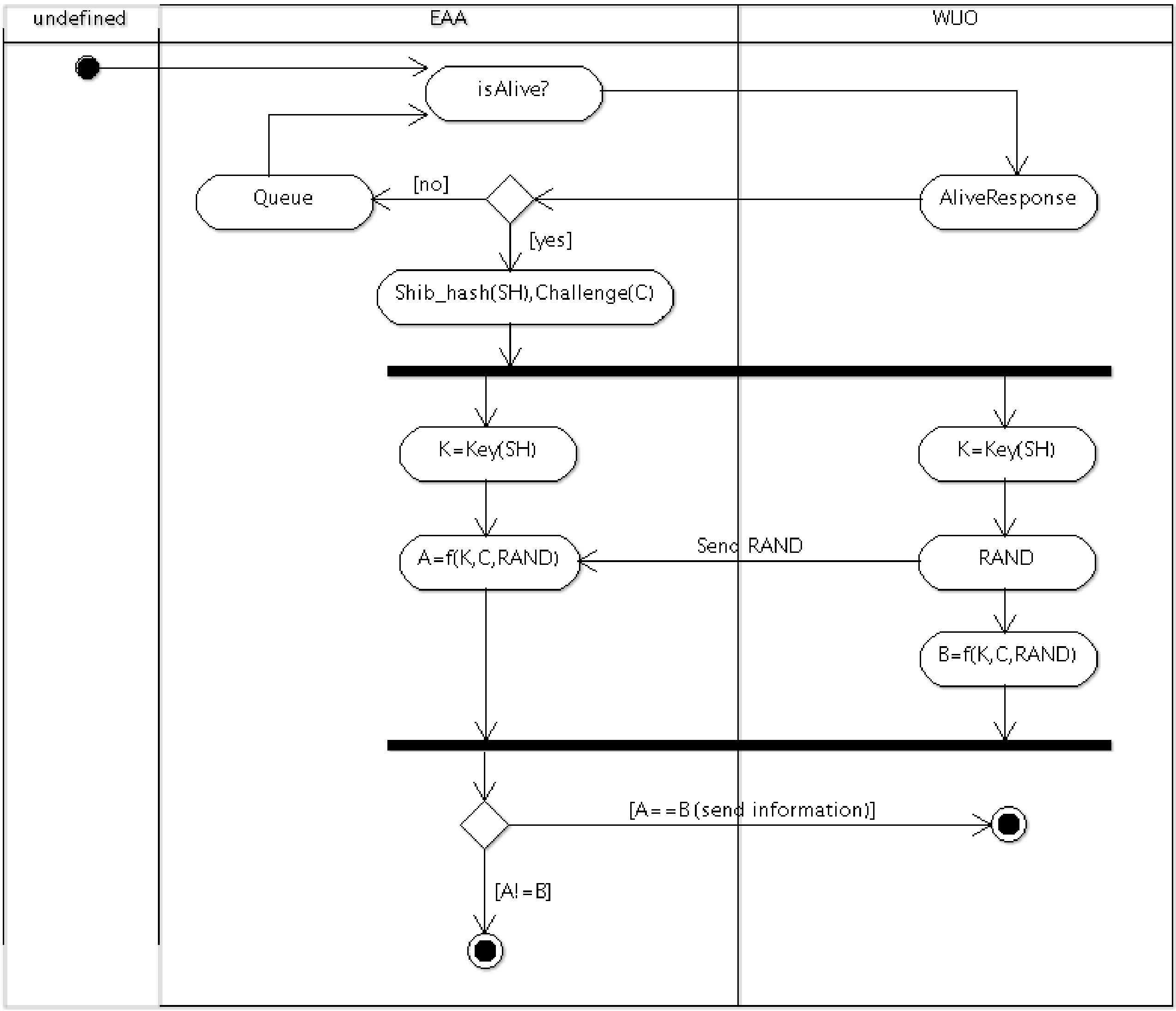


Illustration 14: Activity Diagram explaining the symmetric challenge-response authentication

Explanation:

Shib_hash(SB)	Shibboleth hash used for exact user matching.
Challenge(C)	Random element used as challenge
Key(K)	A key associated with the Shib_hash(SB).
f()	Cryptographic function, e.g. SHA-2
RAND	Random element used as client challenge
A	EAA generated string
B	WUO generated string

Retrieving Information from WUOs

For the case, that no address information is stored at EAA, there must be a way to retrieve this information from a WUO, which already has this information. Considering data security, we can't rely on WUOs opening external channels and should therefor use already existing ways. Each WUO offers a functionality for users to access their account information and we will use exactly this functionality. As the user visits the myAccount page at U3, EAA broadcasts his identification to the WUOs and if one raises a flag, the browser opens via JavaScript a session to that page, retrieves the information from there (remember, the user is already logged in to the federation!) and inserts it into the myAccount form. It's invisible for the user that his information had to be retrieved from a WUO.

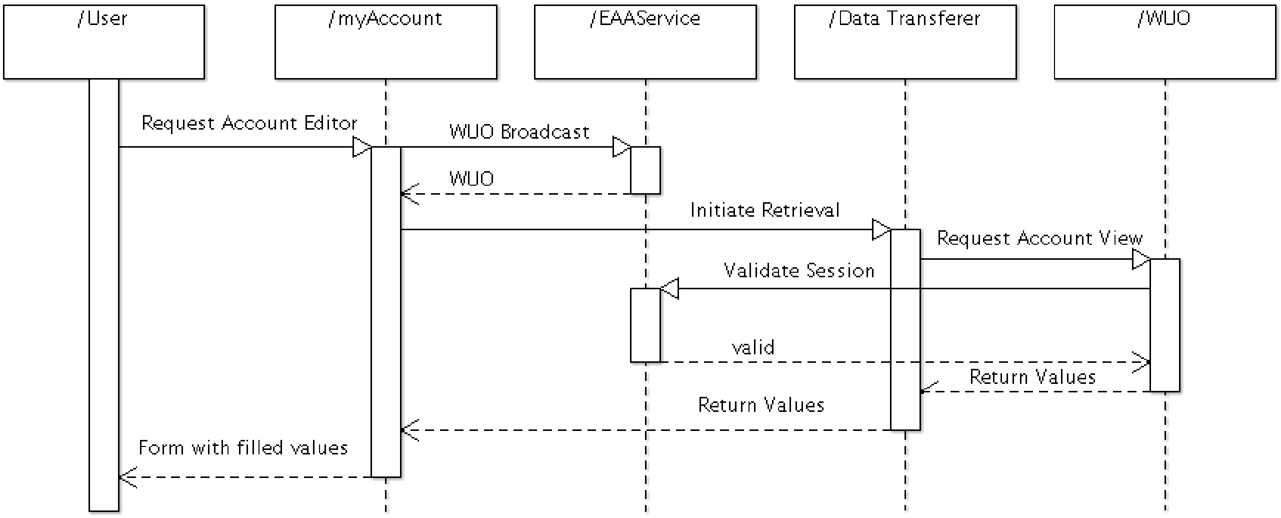


Illustration 15: Sequence Data Transferer

5.4. Components

The EAA system is built of different layers. An application-, a service- and a database-layer. The application layer consists of tools helping with U3 account creation and management. The service layer provides all services to handle Single Sign-On and EAA services. The database layer is divided in a user directory and an affiliation database.

Additionally a JavaScript library is provided to transfer user data from WUO to U3 user creation.

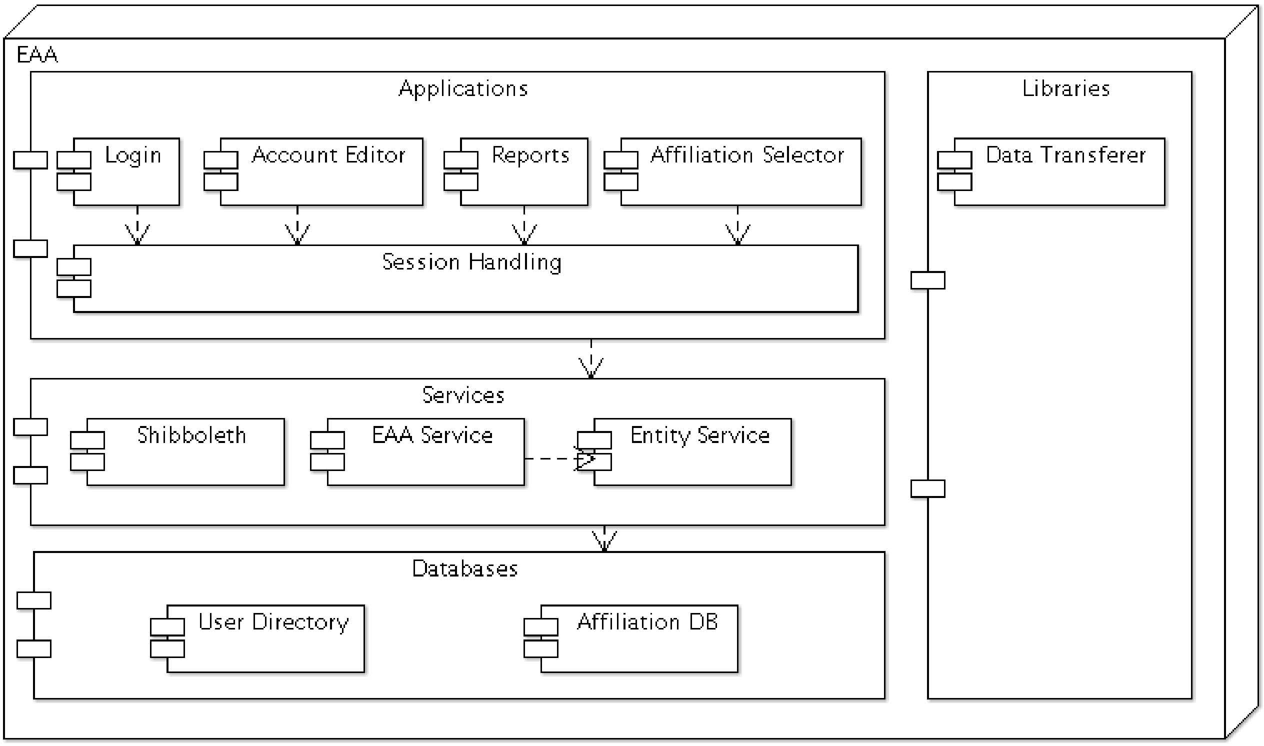


Illustration 16: Components of the EAA system

5.4.1. User Directory

Component	C_EAA1
Name	User Directory
Identifier	C_EAA1
Version	0.1
Stereotype	Directory
Goal	Place for storing account information
Description	This is the central repository to store all EAA accounts. Authentication is done against this directory server.
Input	Query:Text
Output	Results:List
Presentation	
API	LDAP
Actors	Identity Provider

Table 9: Component C_EAA1 User Directory

Some fields should be historicized with a time range of its validity, e.g. email addresses. This will be in any case important when the EAA is used for remote data access.

The Illustration: C_EAA1 Directory structure shows a possible directory structure. Since there is no hierarchical form desired by the requirements it should be kept as flat as possible.

The standard directory is extended with Shibboleth attributes to store information which will be passed to Service Providers.

In certain intervals (e.g. 5 years) users should be requested by the EAA to feedback a sign of life. This allows for disabling dormant users.

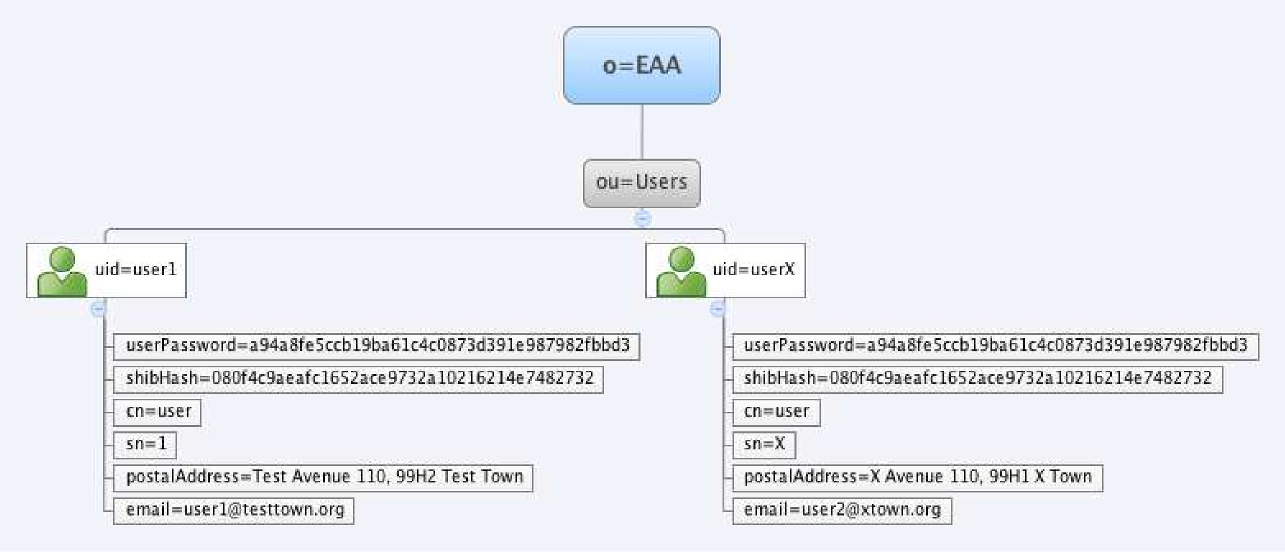

o=EAA
ou=Users
uid=user1
userPassword=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
shibHash=080f4c9aeafc1652ace9732a10216214e7482732
cn=user
sn=1
postalAddress=Test Avenue 110, 99H2 Test Town
email=user1@testtown.org
uid=userX
userPassword=a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
shibHash=080f4c9aeafc1652ace9732a10216214e7482732
cn=user
sn=X
postalAddress=X Avenue 110, 99H1 X Town
email=user2@xtown.org

Illustration 17: C_EAA1 Directory structure

What is LDAP?

- LDAP is an application protocol for querying and modifying data using directory services.
- A directory is a set of objects with attributes organized in a logical and hierarchical manner. A simple example is the telephone directory, which consists of a list of names (of either persons or organizations) organized alphabetically, with each name having an address and phone number associated with it.
- Telecommunication companies introduced the concept of directory services to information technology and computer networking, since their understanding of directory requirements was well-developed after some 70 years of producing and managing telephone directories.

What is a RDBMS

- A relational database management system (RDBMS) is a database management system that is based on the relational model as introduced by E. F. Codd.
- A short definition of an RDBMS may be a DBMS in which data is stored in the form of tables and the relationship among the data is also stored in the form of tables.

Why LDAP over RDBMS?

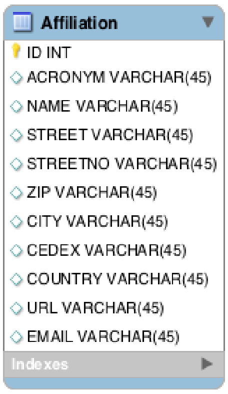
- An LDAP server is a fast, efficient and scalable way to store, manage, search and retrieve data on lots of moderately complex objects.
- Data in an LDAP server can be efficiently replicated, so that multiple servers can participate in providing an LDAP service
- Good application support. Many products and open source solutions have built in LDAP support, and so use of LDAP simplifies application integration.
- Authentication systems, such as PKI (Public Key Infrastructure) generally use LDAP for information storage. Using LDAP enables us to efficiently integrate and make use of external authentication technologies.
- Existing Single Sign-On solutions tend to work better with a directory server then with a RDBMS.

5.4.2. Affiliation DB

Component	C_EAA2
Name	Affiliation DB
Identifier	C_EAA2
Version	0.1
Stereotype	Database
Goal	Store affiliation names
Description	A central database with address information about affiliations. It should be taken into account that the information should be properly split into its components (e.g. street and house number separately) that different output formats can be generated via templates
Input	Query:Text
Output	Results:List
Presentation	none
API	SQL
Actors	User

Table 10: Component C_EAA2 Affiliation DB

There is no accepted standard for the definition of the address of an affiliation. The only way out is to define a pan-European de-facto standard within the EAA. There might be legal restrictions within the single participating countries. The participating facilities agree to find out at home if there are legal confidentiality restrictions for passing affiliation information. Because of national specificities of EU support (e.g. reimbursing) there may be the need for more than one affiliation per user (e.g. a user may be eligible for reimbursement at one affiliation but not at other). Here, however, the proposal is not to deal with this issue within EAA but instead leave this to the local WUOs.



Affiliation

ID INT
ACRONYM VARCHAR(45)
NAME VARCHAR(45)
STREET VARCHAR(45)
STREETNO VARCHAR(45)
ZIP VARCHAR(45)
CITY VARCHAR(45)
CEDEX VARCHAR(45)
COUNTRY VARCHAR(45)
URL VARCHAR(45)
EMAIL VARCHAR(45)
Indexes

Illustration 18: Affiliation Database Table

5.4.3. EAA Service

Component	C_EAA3
Name	EAA Service
Identifier	C_EAA3
Version	0.1
Stereotype	Service
Goal	Deliver the EAA system with functionality and logic
Description	Deliver all functionality needed by the EAA system and abstract from Affiliation DB and User Directory.
Input	See Class Diagram of EAA Service
Output	See Class Diagram of EAA Service
Presentation	none
API	RMI, SOAP, REST, IIOP
Actors	User, Coach

Table 11: Component C_EAA3 EAA Service

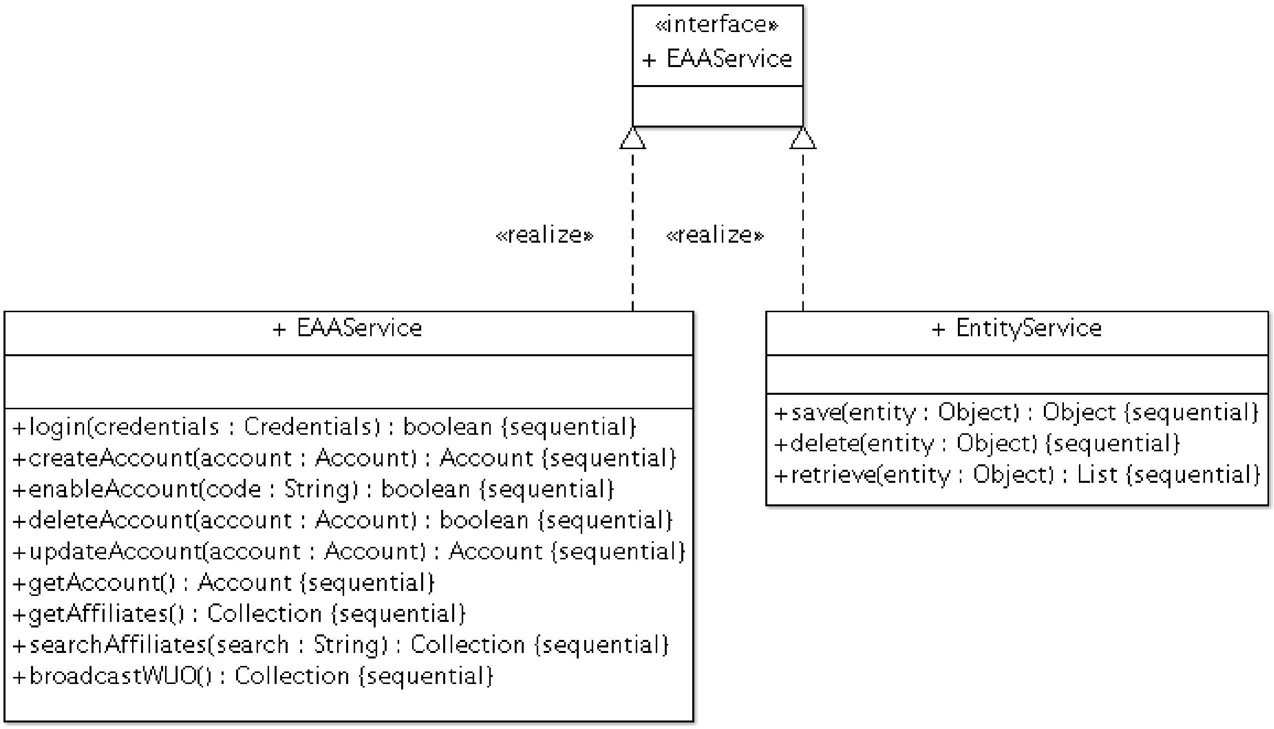

«interface»
+ EAAService
«realize»
«realize»
+ EAAService
+login(credentials : Credentials) : boolean {sequential}
+createAccount(account : Account) : Account {sequential}
+enableAccount(code : String) : boolean {sequential}
+deleteAccount(account : Account) : boolean {sequential}
+updateAccount(account : Account) : Account {sequential}
+getAccount() : Account {sequential}
+getAffiliates() : Collection {sequential}
+searchAffiliates(search : String) : Collection {sequential}
+broadcastWUO() : Collection {sequential}
+ EntityService
+save(entity : Object) : Object {sequential}
+delete(entity : Object) {sequential}
+retrieve(entity : Object) : List {sequential}

Illustration 19: Class Diagram of EAA Service

5.4.4. Shibboleth

Component	**C_EAA4**
Name	Shibboleth
Identifier	C_EAA4
Version	0.1
Stereotype	Infrastructure
Goal	Federated Single Sign-on
Description	Federated Single Sign-on infrastructure based on SAML2. Broadly used and developed.
Input	-
Output	-
Presentation	-
API	HTTP
Actors	User, identity provider, service provider

Table 12: Component C_EAA4 Shibboleth

What is Shibboleth?

Shibboleth is an Internet2 Middleware Initiative project that has created an architecture and open-source implementation for federated identity-based authentication and authorization infrastructure based on Security Assertion Markup Language (SAML). Federated identity allows for information about users in one security domain to be provided to other organizations in a federation. This allows for cross-domain single sign-on and removes the need for content providers to maintain user names and passwords. Identity providers (IdPs) supply user information, while service providers (SPs) consume this information and get access to secure content.

Why use Shibboleth?

- It‘s free and open source: http://shibboleth.internet2.edu
- Built on SAML:
    - XML-based framework for communicating user authentication, entitlement, and attribute information
    - State of the art, designed by industry experts: EMC, HP, IBM, Microsoft, Nokia, Oracle, SAP, Boeing et. al.
    - Considers user privacy a first-order priority.
- Used by national federations in the (higher) education sector (e.g. UK, Switzerland)
- Extending federations across nation boundaries is being worked on.
- Authentication is a dynamic field with ever changing technologies. Using a standard allows us to stay at the cutting edge with a minimum of development effort.

How the Umbrella is built

- The plan is to use the widely tested and used Shibboleth software as the single sign-on component and to build the umbrella on top of it.
- The umbrella consist of custom tailored web-applications, handling all the current use-cases and is extensible for future use-cases to come.

Work for the community

- Run an IdP (Ensure high availability -- downside of „central“ accounts).
- Run the Umbrella portal:
  - Develop and install Umbrella portal on the SP.
  - Run a SP for the Umbrella portal.
  - Support portal users: Run a helpdesk service (consider time zones).
- Support facilities/institutes in joining the federation.
  - Support system administrators setting up SP software.
  - Support system administrators connecting to the IdP.
- Centrally defined and managed affiliation database
  - Keep names and addresses up to date.
- Handling duplicate user accounts.

Work for facilities/institutes

- Train the local user office staff on the new system.
- Install the SP software.
- Adapt the local software (e.g. PSI DUO, SMIS) to use the SP.
  - Allow for dual-login (local as well as Umbrella accounts).
  - Add Umbrella ID attribute to local user DB.
  - Add user management functions to link existing accounts to Umbrella IDs.
  - Add workflow to either a) create new (local) user account on the fly or b) link to existing account if an Umbrella ID is not found in the local user DB.

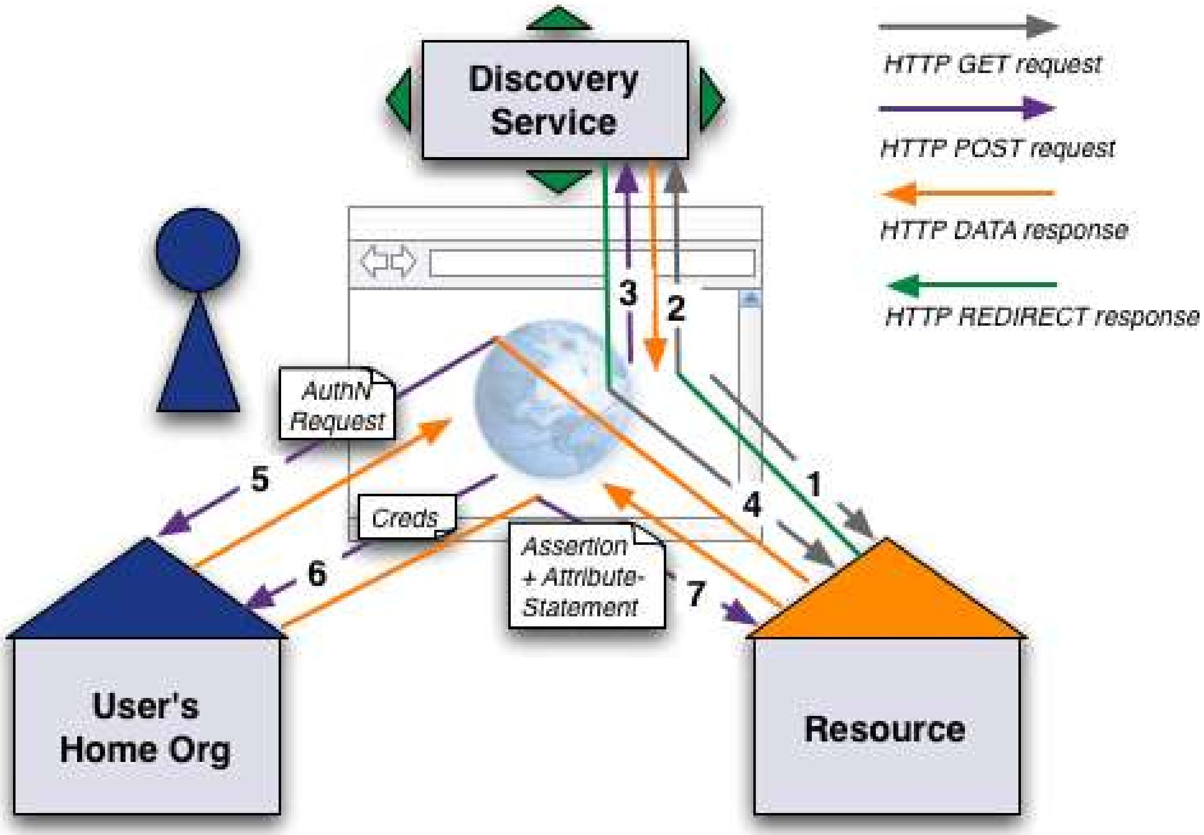


Illustration 20: How Shibboleth works

Step 1:
When you clicked on the 'demo resource' link, your web browser sent an HTTP request to 'aai-demo.switch.ch' for the webpage '/secure/'. The web server answered with an HTTP Redirect to the Discovery Service located at 'wayf-test.switch.ch' because you didn't have yet a valid Shibboleth session because you were not yet authenticated.

Step 2:
After the redirect, the Discovery Service sends your web browser an HTML web page with the pop-up list with all Home Organizations that are available. In this demo, you will only see Home Organizations of our test infrastructure.

Step 3:
The user submits the Home Organization selection form. The Discovery Service answers with a redirect to the session initiator of the resource.

Step 4:
The session initiator creates an authentication request for the chosen Home Organization and submits it through the users browser to the Home Organization.

Step 5:
The Home Organization evaluates the authentication request and answers with the login

page.

Step 6:

The user provides his credentials to the Home Organization. The credentials are checked and an assertion including the users attributes is created according to the attribute filter rules.

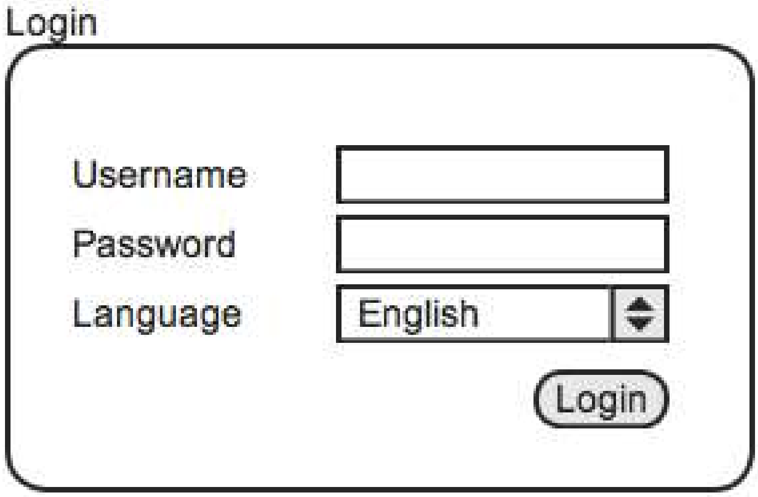
Step 7:

The assertion then is submitted through the users browser back to the resource. Due to the user's attributes, the resource can perform the authorization checks. If the authorization is successful, the user is redirected to the resource page that was initially requested.

5.4.5. Login

Component	**C_EAA5**
Name	Login
Identifier	C_EAA5
Version	0.1
Stereotype	Web component
Goal	Deliver a login mask to authenticate against EAA
Description	Central login mask at EAA. Can be used by WUOs for login dispatching.
Input	Credentials
Output	Session
Presentation	• Login mask
API	HTTP
Actors	User

Table 13: Component C_EAA5 Login


Login

Username

Password

Language: English

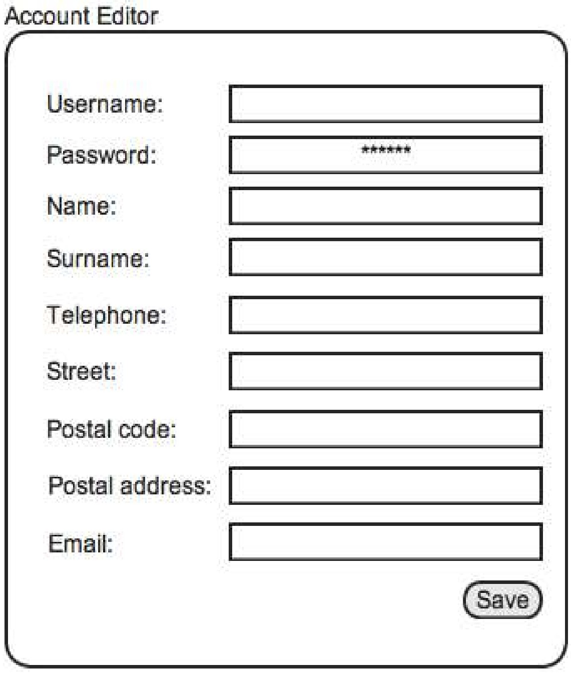
Illustration 21: Web Component: Login Mask

This Component is delivered by the Shibboleth distribution and must be adjusted to meet CI/CD requirements.

5.4.6. Account Editor

Component	**C_EAA6**
Name	Account Editor
Identifier	C_EAA6
Version	0.1
Stereotype	Web component
Goal	Form to change address information
Description	A user can have a look at his information and when needed change them. This information is then disseminated to the WUOs that it can be integrated into their account database.
Input	User Information
Output	Information Broadcast
Presentation	- Account Editor
API	HTTP
Actors	User

Table 14: Component C_EAA6 Account Editor



Account Editor

Username:

Password: ******

Name:

Surname:

Telephone:

Street:

Postal code:

Postal address:

Email:

Illustration 22: Component C_EAA6 Account Editor

5.4.7. Reports

Component	C_EAA7
Name	Reports
Identifier	C_EAA7
Version	0.1
Stereotype	Web component
Goal	Deliver operators of EAA with statistics about its usage.
Description	On a regular basis the EAA system delivers reports to selected people. The specific reports depend on the set of information which is stored at EAA, but some basic reports should contain number of users and login frequency in general. This should be used for statistical evaluations of the use of EAA. An existing reporting tool should be used.
Input	Logged information
Output	Fully formatted reports as PDF
Presentation	none
API	HTTP
Actors	EAA

Table 15: Component C_EAA7 Reports

5.4.8. Affiliation Selector

Component	**C_EAA8**
Name	Affiliation Selector
Identifier	C_EAA8
Version	0.1
Stereotype	Web component
Goal	User selects his affiliation
Description	When changing addresses the user can also supply his affiliation. This is represented as an own component, from which the user can search or browse for finding his/her affiliation. This information is then forwarded to the WUOs together with address information.
Input	Search
Output	Affiliation
Presentation	- Affiliation Selector
API	HTTP
Actors	User

Table 16: Component C_EAA8 Affiliation Selector

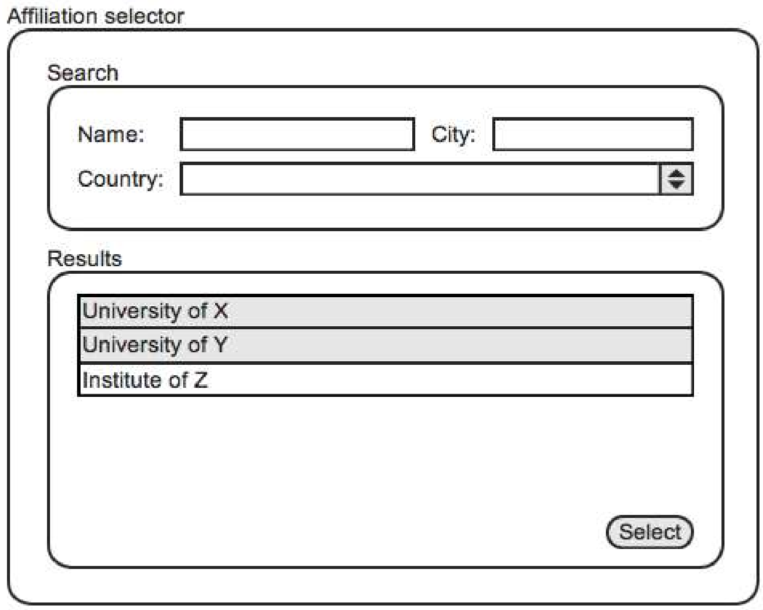


Illustration 23: Component C_EAA8 Affiliation Selector

5.4.9. Data Transferer

Component	**C_EAA9**
Name	Data Transfer
Identifier	C_EAA9
Version	0.1
Stereotype	Web library
Goal	Retrieve address information about a logged-in user from the WUO he/she is registered at.
Description	The data transferer is a library resident in the browser and loaded at the Account Editor. It can be used to transparently dispatch a request from the browser to the WUO to retrieve to local account view and extracts the relevant information from there to be used at EAA.
Input	
Output	Address information
Presentation	none
API	HTTP, JavaScript
Actors	User, Browser

Table 17: Component C_EAA9 Data Transferer

5.4.10. Session Handling

Component	**C_EAA10**
Name	Session Handling
Identifier	C_EAA10
Version	0.1
Stereotype	Web infrastructure
Goal	Keep at user logged-in at EAA
Description	The session handling is used for all EAA services. After login, a session is established and used to identify this user over all his requests to the EAA. After an amount of inactive time, the session is then destroyed at the server. It is not the same session as the Shibboleth session used for SSO!
Input	Login
Output	Session
Presentation	none
API	HTTP
Actors	User

Table 18: Component C_EAA10 Session Handling

5.4.11. Entity Service

Component	C_EAA11
Name	Entity Service
Identifier	C_EAA11
Version	0.1
Stereotype	Entity
Goal	Deliver core entity system with additional entities used by EAA
Description	The only entity used in EAA is a representation class for affiliations. It is used mainly for transfering the list of affiliations.
Input	See Illustration Class Diagram of EAA Entities
Output	See Illustration Class Diagram of EAA Entities
Presentation	none
API	RMI
Actors	EAA Service

Table 19: Component C_EAA11 Entity Service

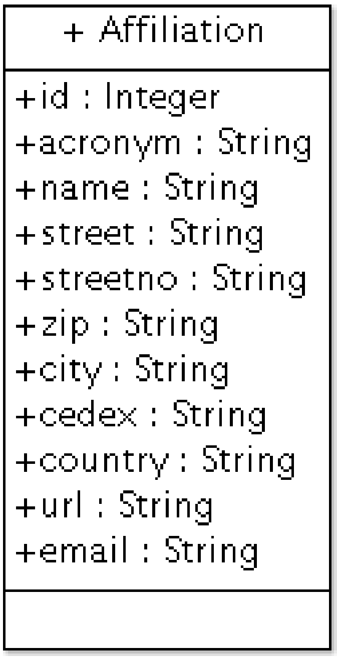


Illustration 24: Class Diagram of EAA Entities

6. EUU, EuroFEL User Umbrella

Function of the Umbrella

Once an EAA (European Authentication and Authorization) scheme is established, a multitude of services can be offered by the UUU (Unified User Umbrella) to the community, ranging from access to experimental and published data stored at remote facilities and proposal handling to PR and science-political issues. The EUU (EuroFEL User Umbrella) as the first version concentrates on proposal-related issues.

Proposals consist of two parts, a general part, which focuses on the scientific aspects of the proposed experiment and is largely facility-independent and a local part, which describes the facility-specific aspects.

For the foreseeable future, there will be a coexistence of UUU and non-UUU proposals and the workflows of UUU and local WUOs have to be designed such that they are able to handle both options. The duty of the EUU is, therefore, to optimize/harmonize the generation of the general part and to forward the proposer to the respective local facility for entering the local part of the proposal (umbrella concept).

Standard for the General Part

As part of the EUU, a template will be defined, which specifies the various items (e.g. topics, sequence, length) for the general part. For the – predominantly – iterative-type experiments this will allow users to prepare a new proposal based on a previous proposal in an efficient way. In addition to the present context such a standard would also be highly welcomed by the referees who have to rate and rank tens to hundreds of proposals during a proposal session. To agree on a pan-European template is mainly a political issue between the various facilities and WP2 will set out to push for such a standard.

Proposal Submission

A user registered at the EUU is allowed to submit a proposal for an experiment at any of the participating facilities. For that, he/she logs on at the EUU and selects from a menu the facility of choice. He/she enters the general part and after that the local part of the proposal. After that, the standard proposal handling at the facility continues. The proposal is stored within the local WUO with read access only to the proposers and the management of the local facility.

Proposal Export

Proposal information is confidential. Access is allowed only to the proposer and the co-proposers specified in the document and to the management of the respective facility. Proposers have the right to access their own proposal, e.g. as basis for submitting a follow-on proposal. EUU in handshake with the local WUOs will provide the corresponding export mechanism.

Proposal resubmission

Quite often users resubmit a modified version of a proposal to the original or another facility. For that, they can access the old proposal, modify it and submit it to the facility. Two options are possible, which have still to be decided upon:

- Manual: the user logs on the facility where the proposal is stored and generates a copy. Then, if the proposal is to be submitted to another than the home facility, he/she logs on that facility, which because of SSO does not need any identification. After performing the modifications based on the copy generated, the proposal is submitted.
- EUU-supported: the user logs on the 'old' facility, fetches the 'old' proposal and selects the 'new' facility. After that, the EUU transfers the user to the new facility and submission proceeds as usual.

In any case, a user can access an 'old' proposal only if he/she is author or co-author.

Access to Services at a Local WUO

In the first version, central user services will concentrate on proposal-related issues. For further services, users are forwarded to the local facility WUO's.

6.1. Thoughts behind

- With this approach we can keep up a maximum level of confidentiality while at the same time have a maximum of flexibility.
- Because of the high confidentiality risk no proposals should be stored at the U3.
- Storage of the proposals is under full control of the users and facilities.
- Exporting proposals from WUOs could be a data security issue which must be treated accordingly.
- If proposals would not be exportable from the WUOs the users would have to store and organize them locally by themselves, forcing them to archive and retrieve proposals by themselves. In addition, the proposal format could have changed between archiving and reuse of a proposal so that the structure could be outdated.
- The export mechanism allows addition of new functionality with minimum effort.
- This approach, however, requires the agreement on a formal definition of the structure of the proposal. A proposal should be subdivided into two parts: a general exportable description of an experiment and a facility-specific description of infrastructure involved.

6.2. Use Cases

In the following a set of functions is defined which the EUU system has to provide. The use cases describe only the expected functionalities but they do not reveal any technical constraints. The actor WUO is a technical actor only.

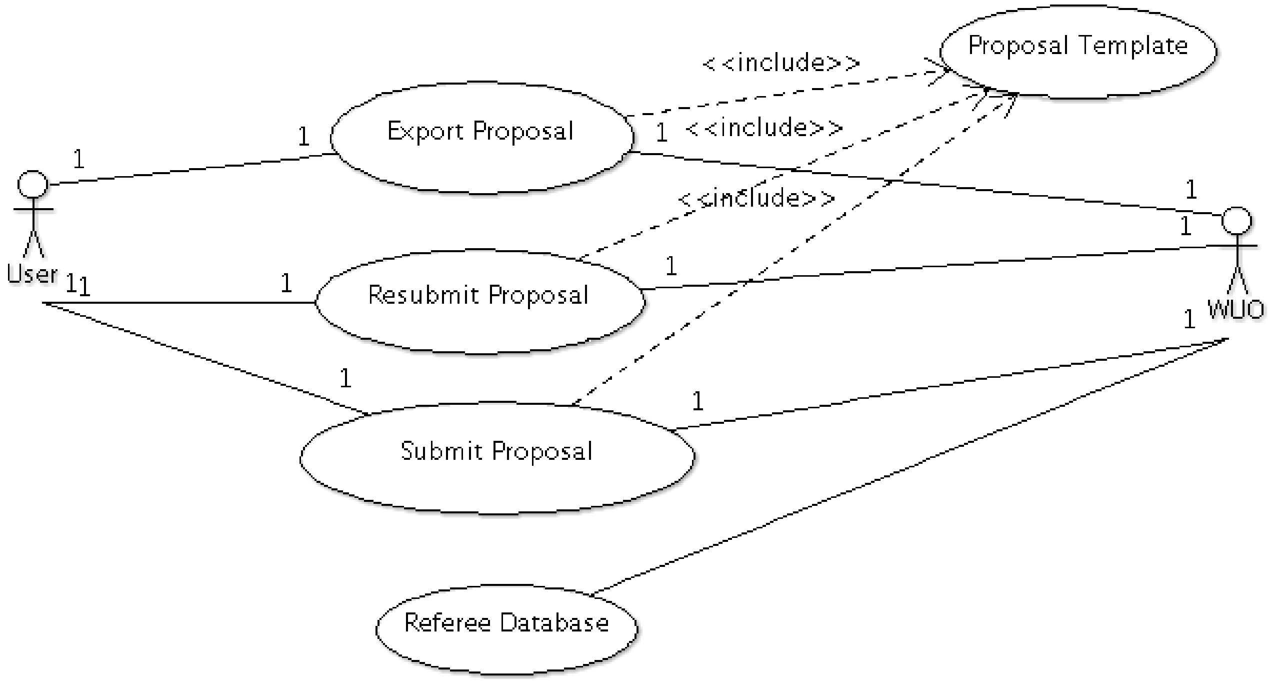


Illustration 25: Use Cases EUU

Actors	EUU
WUO	Web User Office able to export and import proposals.
User	U3 User

Table 20: Actors EUU

6.2.1. Submit Proposal

Use Case	UC_EUU1
Name	Submit Proposal
Identifier	UC_EUU1
Version	0.1
Goal	Submit a proposal to a facility
Summary	A proposal is submitted by a user to a facility.
Actors	User, WUO
Stakeholders	User, WUO
Preconditions	• User must exist • User must be logged-in at U3 • User must exist at WUO
Triggers	User manually selects to submit a proposal.
Course of events	1. User clicks on “Submit Proposal” 2. User either uploads or fill out the proposal 3. User selects facility 4. User is forwarded to facility with the general part of the proposal to fill out the facility-specific part.

Table 21: Use Case UC_EUU1

6.2.2. Export Proposal

Use Case	UC_EUU2
Name	Export Proposal
Identifier	UC_EUU2
Version	0.1
Goal	Export a Proposal from a facility
Summary	The general part of a submission can be exported while the user is visiting a WUO. This information is exported as a XML file.
Actors	User, WUO
Stakeholders	User
Preconditions	• User must be logged-in at WUO • User must be author or co-author of the proposal
Triggers	User manually triggers the download of a proposal file
Course of events	1. User logs in at WUO 2. User selects proposal to export 3. User clicks on 'Export Proposal' 4. User selects place to save

Table 22: Use Case UC_EUU2

6.2.3. Resubmit Proposal

Use Case	**UC_EUU3**
Name	Resubmit Proposal
Identifier	UC_EUU3
Version	0.1
Goal	Resubmit a proposal
Summary	Reuse an already submitted proposal for a new proposal. This use case utilizes both UC_EUU1 and UC_EUU2.
Actors	User, WUO
Stakeholders	User
Preconditions	• User must exist both at U3 and WUO • User must be logged in at U3 and WUO
Triggers	User exports an existing proposal and uses it to submit a new proposal
Course of events	Composition of UC_EUU2 and UC_EUU1

Table 23: Use Case UC_EUU3

6.2.4. Proposal Template

Use Case	UC_EUU4
Name	Proposal Template
Identifier	UC_EUU4
Version	0.1
Goal	Define a XML-Schema to describe the general part of a proposal
Summary	A XML schema must be defined to hold the general part of a proposal. This schema is then used to import and export proposals from and to WUOs
Actors	-
Stakeholders	User, WUO
Preconditions	XSD must exist
Triggers	-
Course of events	-

Table 24: Use Case UC_EUU4

In the first step a survey of the existing formats will be made followed by the proposal of a compromise format.

6.2.5. Referee Database

Use Case	**UC_EUU5**
Name	Referee Database
Identifier	UC_EUU5
Version	0.1
Goal	Have a database where all referees are contained
Summary	Have a central list of referees. This part is very security relevant and we are not sure if it is safe to provide this information! Therefor we mention the need but a detailed specification will be delayed until agreement between the facility managers is reached.
Actors	WUO
Stakeholders	WUO
Preconditions	
Triggers	
Course of events	

Table 25: Use Case UC_EUU5 Referee database

6.3. *Specification*

6.3.1. General

The first Version of the EUU consists mainly of the proposal submission part. Since the EUU application map is meant to be extended over time, we have to keep extensibility in mind. Functional modules should be addable easily. This means that we have to select a web application framework or PaaS (Platform as a Service) and to build an infrastructure to support this need.

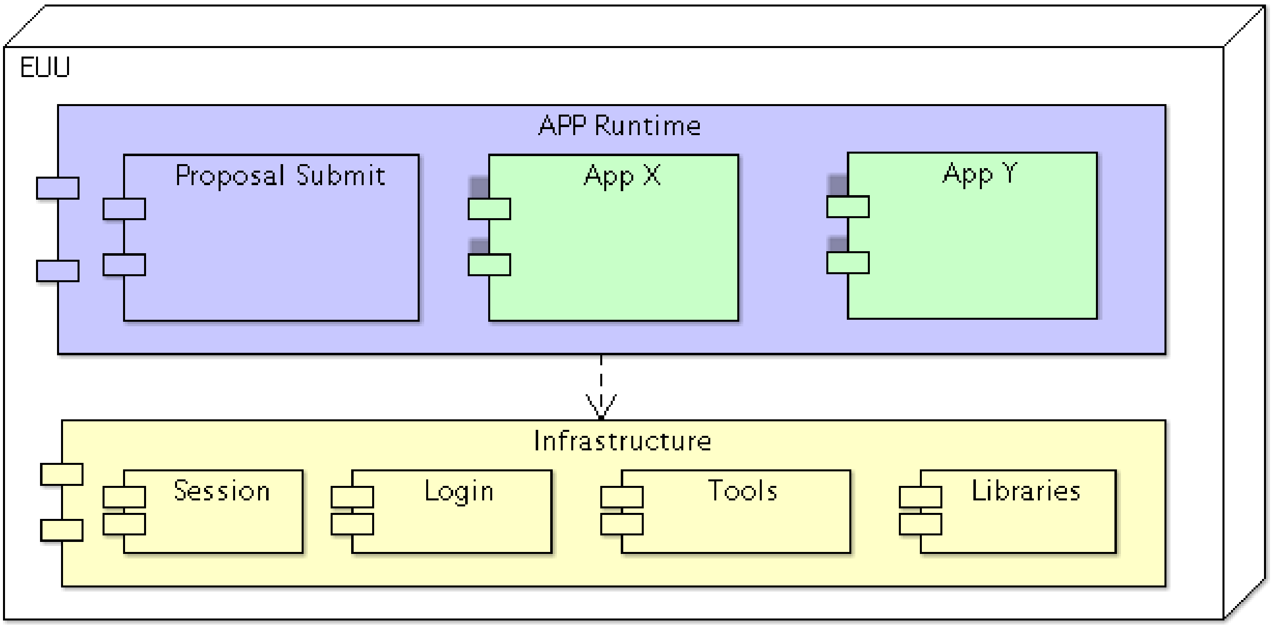


Illustration 26: Extendability of EUU

6.3.2. Sequence

Following sequence diagram shows the export of a proposal from a WUO to a Users computer.

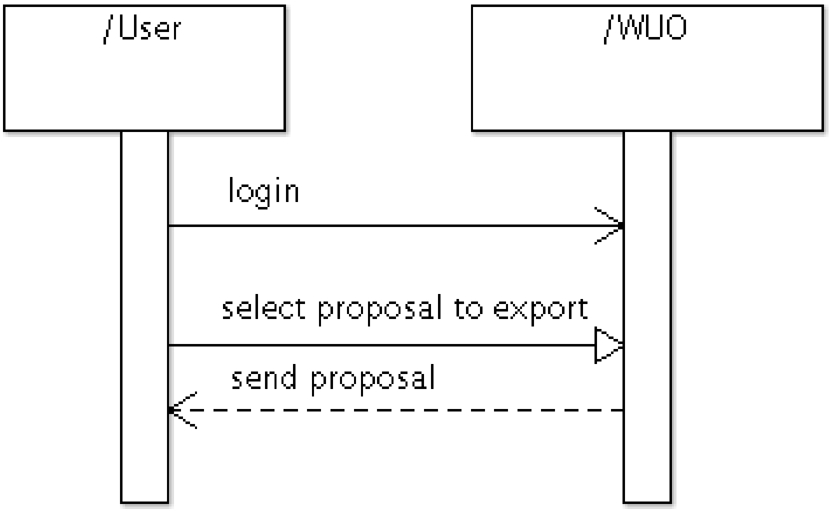


Illustration 27: Proposal export as sequence diagram

For this to happen most transparently to the user, WUOs must be extended to support the EUU proposal definition. Currently, there is a large variety of formats accepted at the various WUOs, ranging from plain text over DOC, TEX, RTF to PDF, in one case PDF together with RTF. Whereas PDF clearly allows the 'nicest' layout, users can not use / edit such a document for resubmitting a proposal, what would be a big help for users in view of the predominantly iterative character of the experiments. Therefore, a scheme is developed and proposed, which allows a maximum of flexibility for submitting an edited version of the general part for the same or another participating facility. Changes to be done at WUOs are as follows:

- Case 1: WUOs already have an internal representation of a proposal. Usually only a transformation step is needed to change the specific WUO representation to the U3 representation.
- Case 2: WUOs might have an existing export function. If this is the case, just another download button is needed for the U3 representation. Export functionality must be provided if it doesn't already exist.

This sequence diagram explains what happens at the WUO under the hood:

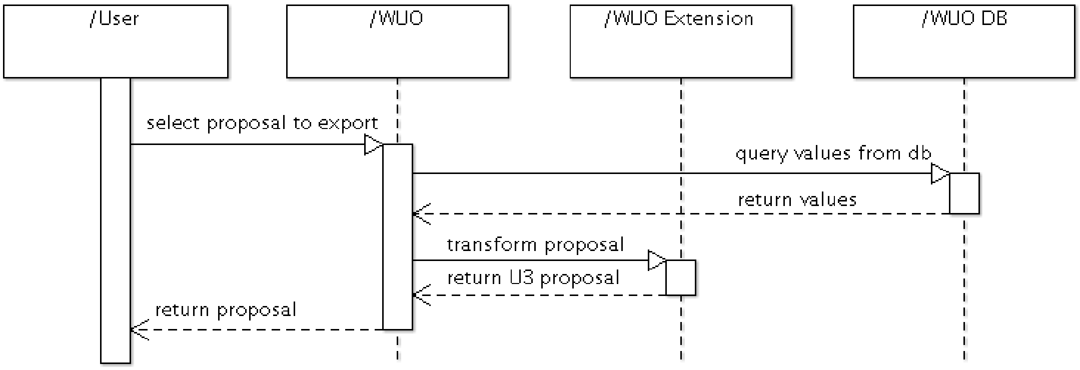


Illustration 28: WUO actions during proposal export as sequence diagram

Following a sequence diagram showing what happens during proposal submission

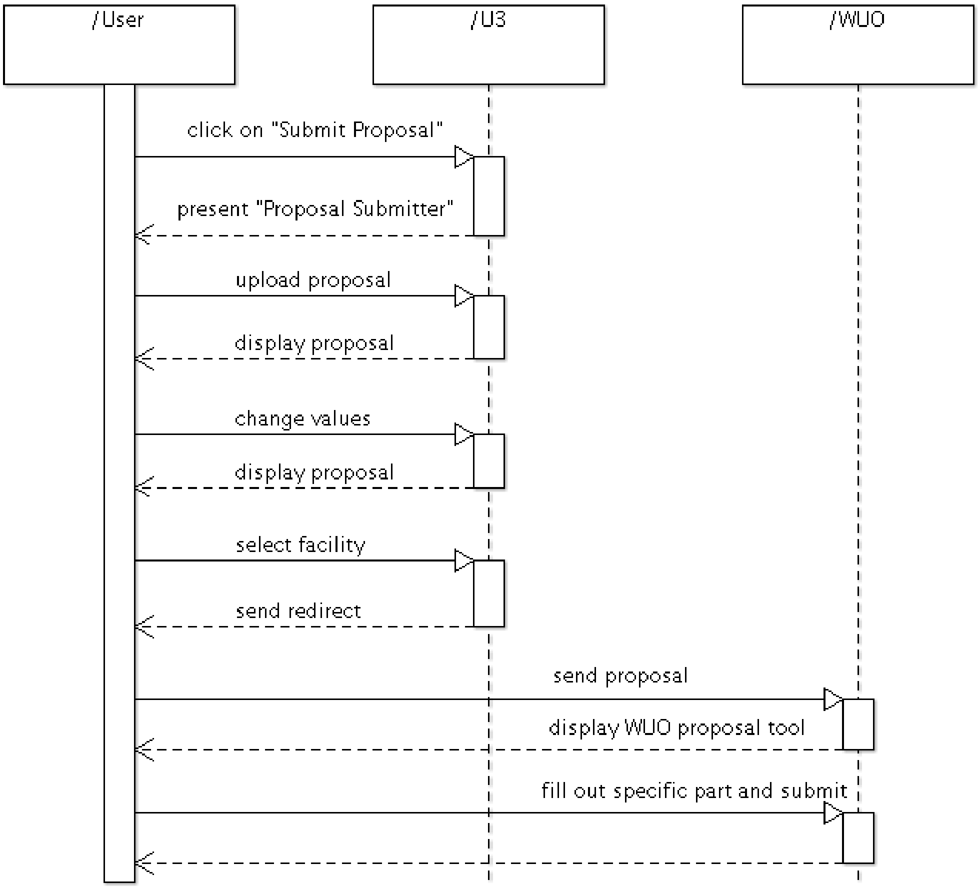


Illustration 29: Proposal submission as sequence diagram

A user with proper authorization (e.g. being one of the original proposers) can upload a proposal to the proposal submitter. The content is then displayed and further editing of the general part is possible. If no proposal is uploaded, the empty editor is displayed for editing. When the user has finished filling out his proposal he/she selects a facility where he/she wants his experiment to be performed. The user is forwarded to this facility together with the general part of the proposal and then adds the facility-specific part and completes and submits the proposal.

To fulfill the needs of the proposal submission process there are extensions to be implemented at the WUOs:

- An infrastructure is needed to accept incoming XML files containing the general part of a proposal.
- A transformation step is needed to change the U3 format to the internal representation of a proposal.
- PDF to text (+ figures) back-conversion

6.4. Components

Following an overview of the components needed for the proposal submission.

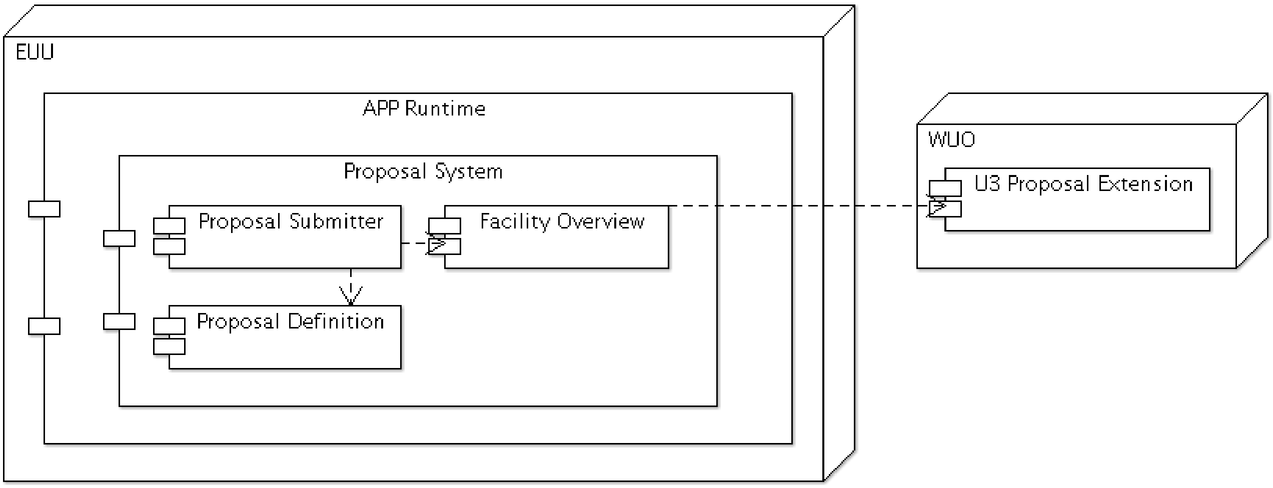


Illustration 30: Components EAA

For easy extendability of the EUU an application container or framework must be selected to fulfill the need of reduced complexity in extending the EUU with further applications. The proposal system itself is one of those applications and consists of an editable representation of the general part of a proposal as well as its definition. A facility overview is needed for dissemination of proposals to WUOs.

An integration layer is needed at the WUOs which supports the functionality to bidirectionally transform the U3 proposals into a WUO internal representation of them and further to import and export them. The export should happen as a download of the proposal in the U3 proposal format to his computers hard disc. The import should happen as an import to the WUOs internal proposal editor and provide the general fields of the web form filled out with the data from the proposal submitter.

6.4.1. APP Runtime

Component	**C_EUU1**
Name	APP Runtime
Identifier	C_EUU1
Version	0.1
Stereotype	Runtime
Goal	Provide a runtime for EUU applications.
Description	For eased development there is a runtime which provides out-of-the-box API and deployment functionality.
Input	-
Output	-
Presentation	-
API	HTTP
Actors	U3

Table 26: Component C_EUU1 APP Runtime

Conceptually the APP Runtime should be similar to existing runtimes like Google App Engine. Preferably an open source framework should be selected to enable “Platform as a Service” functionality.

6.4.2. Proposal Submitter

Component	C_EUU2
Name	Proposal Submitter
Identifier	C_EUU2
Version	0.1
Stereotype	Web component
Goal	Handle creating and submitting the general part of a proposal to WUOs
Description	This web component allows the import of proposals in U3 format. It displays the data in a form and allows editing of the data itself. The correctness of the data can be verified before selecting the WUO to be submitted to.
Input	XML file : U3 format
Output	Proposal submit : HTTP-Request
Presentation	• Proposal submitter
API	HTTP
Actors	User, WUO, U3

Table 27: Component C_EUU2 Proposal Submitter

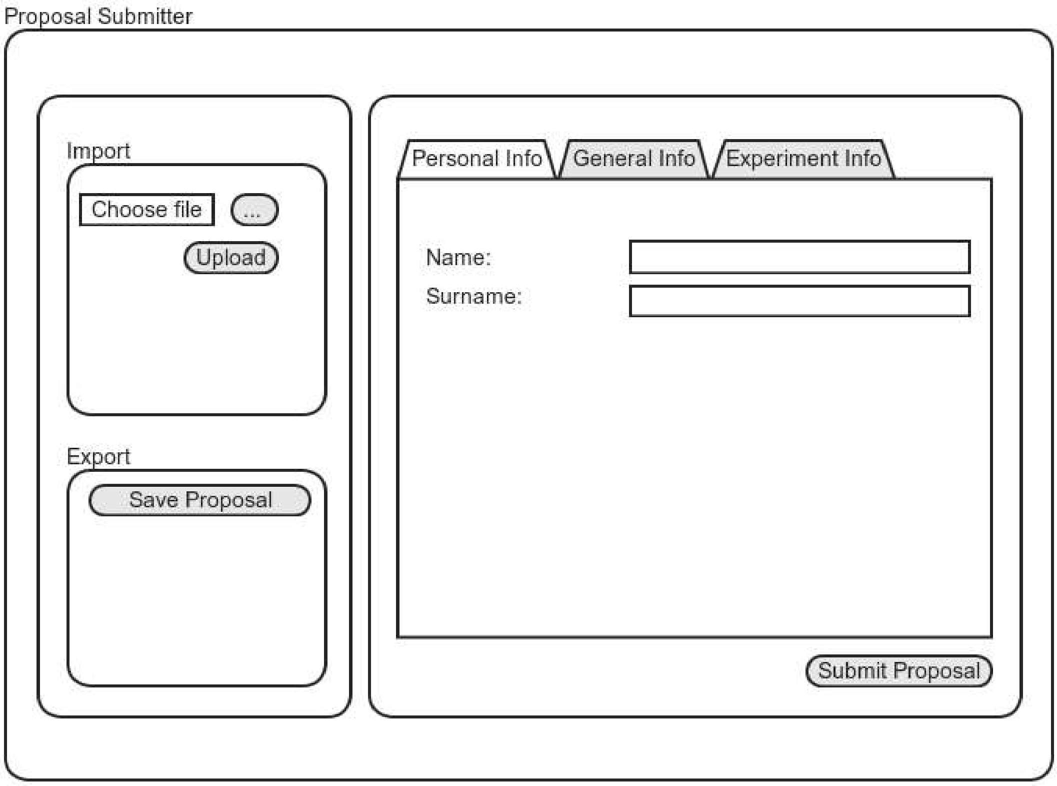


Illustration 31: Component C_EUU2 Proposal submitter

6.4.3. Proposal Definition

Component	C_EUU3
Name	Proposal Definition
Identifier	C_EUU3
Version	0.1
Stereotype	XSD schema
Goal	Create the U3 proposal definition.
Description	Define a XSD schema containing all the relevant information on the general part of a proposal.
Input	-
Output	-
Presentation	-
API	-
Actors	-

Table 28: Component C_EUU3 Proposal Definition

6.4.4. Facility Overview

Component	C_EUU4
Name	Facility Overview
Identifier	C_EUU4
Version	0.1
Stereotype	Web component
Goal	Handle facility selection
Description	This web component displays a list of facilities to which a proposal can be submitted. It allows selection of entries.
Input	Facilities:List
Output	SelectedFacility:Facility
Presentation	• Facility overview
API	HTTP
Actors	User

Table 29: Component C_EUU4 Facility Overview

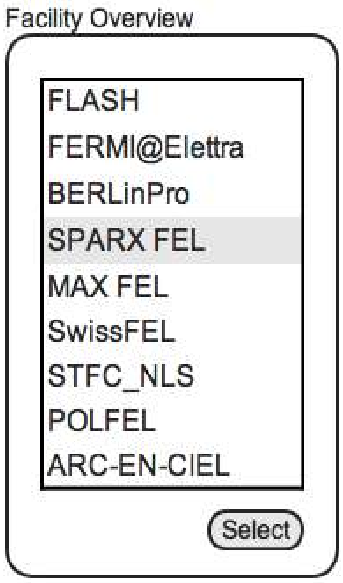


Illustration 32: Component C_EUU4 Facility Overview

6.4.5. U3 Proposal Extension

Component	C_EUU5
Name	U3 Proposal Extension
Identifier	C_EUU5
Version	0.1
Stereotype	Library
Goal	U3-enable the WUO proposal system
Description	This library provides functionality to connect to WUO proposal system to EUU. Integration work is needed at the WUO site to implement import and export as well as bidirectional transformation of proposal definitions.
Input	-
Output	-
Presentation	none
API	
Actors	WUO

Table 30: C_EUU5 U3 Proposal Extension

The U3 Proposal Extension is mainly responsible to export and import proposals from and to WUOs. To integrate the new U3 format into the WUO two XSLT transformations must be implemented to be able to bidirectionally change proposals. The WUOs can still change their internal proposal format and the system would still be working as long as the transformations are updated as well.

The proposal export involves a transformation from internal representation to U3 format and a XML file to download.

The proposal import relies on an extension of the WUO's proposal editing. It should allow the U3 proposal to be sent as XML in a HTTP POST attribute and corresponding transformation from the U3 format into an internal representation and automatic completion of form fields.

A proposal should only be exportable if the logged-in user was the author or co-author of that paper.

Alternatively, an export / import facility for PDF documents should be created since many WUOs already incorporate PDF export. A PDF structure must be defined and agreed on.

7. Coaching

A user enters the respective page of the EUU website. A list of FAQ is provided, which may already solve some of the issues. In case of further need the user can fill a LoI (Letter of Intent). For that, he/she is provided a template with specific questions concerning e.g. science field, sample information, detection technique. The Umbrella Manager directs this LoI to one of the coaches and the coach comments on the LoI, in general electronically. Personal contact to the applicant is up to the decision of the coach.

There may be a second iteration, e.g. with a formal check of a proposal by the user. In any case, however, the full responsibility for the final proposal is with the user.

It will be decisive for the success of this novel service to attract top scientists of the respective scientific fields as coaches. In that respect, a good way has to be found to honor this activity. The idea is to have peer coaches like peer referees. One idea would be to award a coaching diploma signed by e.g. the EuroFEL coordinator which could be cited in the cv.

In order to ease the work of the coaching coordinator - who distributes requests to individual coaches - users should be requested to specify their question by specifying items defined in a category tree.

7.1. Thoughts behind

- There should be an interactive part to answer questions and a static FAQ.
- Increasingly, users are coming to the facilities with no or very low experience in working in such an environment.
- For them, a coaching service is offered, which is aimed at helping them to pass the first threshold barriers.
- This will require experienced coaches
- The coaches must be ‘protected’ against excessive load.
- These coaches are appointed in the first phase by EuroFEL, later by a more general community body (e.g. common ELISA/NMI3).
- They work on a peer basis, like referees.
- There is a limited amount of iterations in a question.
- A question can be anonymized and added to the FAQ part.
- A good way has to be found to honor coaches.
- Define a category tree
- Responsibility lies with the user and we exclude the jurisdiction of a court.

7.2. Use Cases

In the following a set of functions is defined which the coaching system has to provide. The use cases describe only the expected functionalities but they do not reveal any technical constraints. The figure also shows the actors (roles) associated to the functionality. The Coordinator extends the role Coach and inherits all functionality from it.

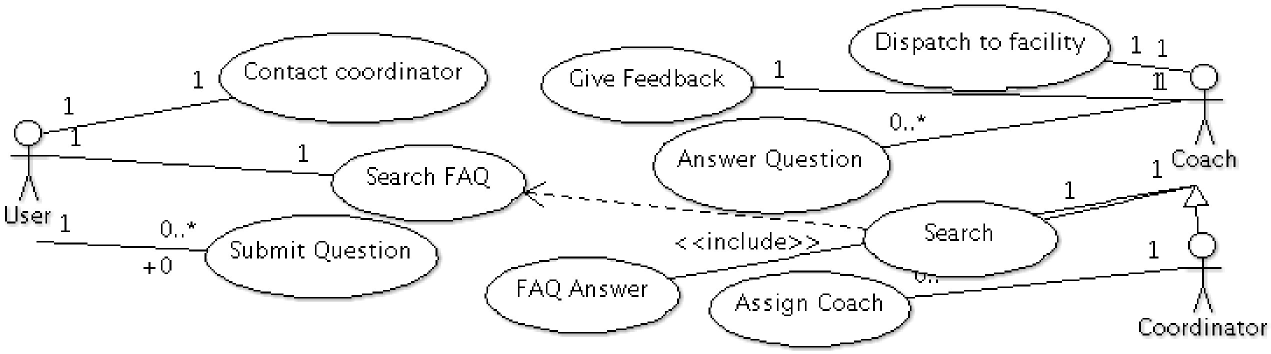


Illustration 33: Use Cases Coaching

Actors	Coaching
User	U3 User
Coach	Assigned to the question of the U3 User. He/She answers the question, can stay anonymously if he/she wants
Coordinator	Dispatches user questions to coaches. Should be as neutral as possible.

Table 31: Actors Coaching

7.2.1. Submit question

Use Case	UC_C1
Name	Submit Question
Identifier	UC_C1
Version	0.1
Goal	Submit a question
Summary	User submits a question regarding a possible experiment at a facility
Actors	User
Stakeholders	User, Community
Preconditions	User must be registered and logged in at the U3 (Umbrella)
Triggers	User clicks on “Pose a Question” on portal
Course of events	1. User clicks on “Pose a Question” 2. User enters his question using a template 3. User submits the question

Table 32: Use Case UC_C1

7.2.2. Assign coach

Use Case	UC_C2
Name	Assign Coach
Identifier	UC_C2
Version	0.1
Goal	Assign a coach to a question posed
Summary	After the user has entered a question a coordinator is responsible for dispatching the question to an expert coach.
Actors	Coordinator
Stakeholders	User, Coach
Preconditions	User has submitted a question which is not yet assigned to a coach
Triggers	User submits a question
Course of events	1. Coordinator gets a list of open questions 2. Coordinator clicks on action “Assign” 3. Coordinator chooses a coach from a list 4. Coordinator assigns the question to coach

Table 33: Use Case UC_C2

7.2.3. Answer question

Use Case	UC_C3
Name	Answer Question
Identifier	UC_C3
Version	0.1
Goal	Question is answered
Summary	Coach and User can have an anonymous conversation. There are up to 3 iterations possible, in order to limit load on the coach. In any case, the responsibility for further actions is always with the user.
Actors	Coach, User
Stakeholders	User, Community
Preconditions	User has submitted a question which is assigned to a coach
Triggers	Coordinator assigns question to coach
Course of events	1. Coach answers an open question 2. User decides if question is answered satisfactorily and either iterates (via step 1 if number of iterations is not exceeded), finishes or contacts the coordinator

Table 34: Use Case UC_C3

7.2.4. FAQ answer

Use Case	UC_C4
Name	FAQ Answer
Identifier	UC_C4
Version	0.1
Goal	Create FAQ-entry from a question
Summary	If a satisfactory answer is given to a question, question and answer(in anonymized form) can be entered into the FAQ database for public use. Additionally a routine should regularly check non FAQ'ed QAs for similarity and propose these candidates to coaches.
Actors	User, Coach
Stakeholders	User
Preconditions	QA iterations finished
Triggers	Finishing the QA triggers a screen to ask the user if she wants to “FAQ” the answer.
Course of events	1. User clicks on “question is answered” 2. The system provides a form containing the anonymized QA 3. Coach inspects the values for FAQ 4. QA is put into FAQ 5. Coordinator and coaches, however, have access to all archived QAs

Table 35: Use Case UC_C4

7.2.5. Search FAQ

Use Case	UC_C5
Name	Search FAQ
Identifier	UC_C5
Version	0.1
Goal	Search FAQ
Summary	Search functionality for the user to find similar questions already posed. The user is allowed to access FAQ's which are defined as publicly available by the authors.
Actors	User
Stakeholders	User, Coach
Preconditions	none
Triggers	User starts a query
Course of events	1. User enters the text in search field 2. User clicks the submit button 3. Elements found are presented

Table 36: Use Case UC_C5

7.2.6. Search

Use Case	UC_C6
Name	Search
Identifier	UC_C6
Version	0.1
Goal	Search in all questions and answers
Summary	Function for coordinator and coaches to search for a specified text or tag in all answers and questions to help find similar questions.
Actors	Coach, Coordinator
Stakeholders	Coach, Coordinator
Preconditions	Coach must be registered and logged in at U3
Triggers	Coach starts a query
Course of events	1. Coach enters a text in search field 2. Coach clicks on submit button 3. Elements found are presented

Table 37: Use Case UC_C6

7.2.7. Dispatch to facility

Use Case	UC_C7
Name	Dispatch to facility
Identifier	UC_C7
Version	0.1
Goal	Forward a user to a beamline manager at a local facility
Summary	If the question is too facility dependent the coach can forward the user to a local facility beamline manager to continue questioning. The further conversation takes place within the local facility infrastructure, so nothing is archived.
Actors	Coach, User, Beamline Manager
Stakeholders	User, Community
Preconditions	Coach must be assigned to question and question must be facility dependent.
Triggers	Coach decides to forward the user to local facility
Course of events	1. Coach decides to forward user to a facility 2. Coach decides which facility 3. User receives a message whom to contact

Table 38: Use Case UC_C7

7.2.8. Contact coordinator

Use Case	UC_C8
Name	Contact Coordinator
Identifier	UC_C8
Version	0.1
Goal	User can contact coordinator to get feedback
Summary	If the user believes his question was not answered properly after the iterations he can contact the coordinator to get feedback concerning this case.
Actors	User, Coordinator
Stakeholders	User
Preconditions	All iterations used up but the question remains unanswered
Triggers	User decides to contact coordinator and clicks on button
Course of events	1. User clicks on “Contact Coordinator” 2. User enters his problem or complaint 3. Coordinator receives report 4. Coordinator “gives feedback” (see UC_C9)

Table 39: Use Case UC_C8

7.2.9. Give feedback

Use Case	UC_C9
Name	Give feedback
Identifier	UC_C9
Version	0.1
Goal	Give feedback to a user
Summary	Coordinators and Coaches
Actors	Coach, User, Coordinator
Stakeholders	User, Community
Preconditions	Coach or Coordinator contacted
Triggers	• Coach forwards the user to local beamline manager • Coordinator was contacted by use
Course of events	1. Coach/Coordinator enters answer text into form 2. Coach/Coordinator clicks on submit 3. Message is submitted to user

Table 40: Use Case UC_C9

7.3. Specification

7.3.1. General

The coaching system consists mainly of a workflow for submitting and answering questions and a FAQ. Following some conditions to meet:

- In order to limit the load, a coach is normally not known to a questioner. On the other hand, however, a coach is free to provide his/her name at any time.
- The procedures for selecting the coordinator and the coaches will be defined in the EuroFEL consortium agreement.

7.3.2. Workflow

Illustration 3 shows the proposed workflow for the coaching system. There are four swimlanes (A, B, C, D) indicating the roles related to the actions and decisions.

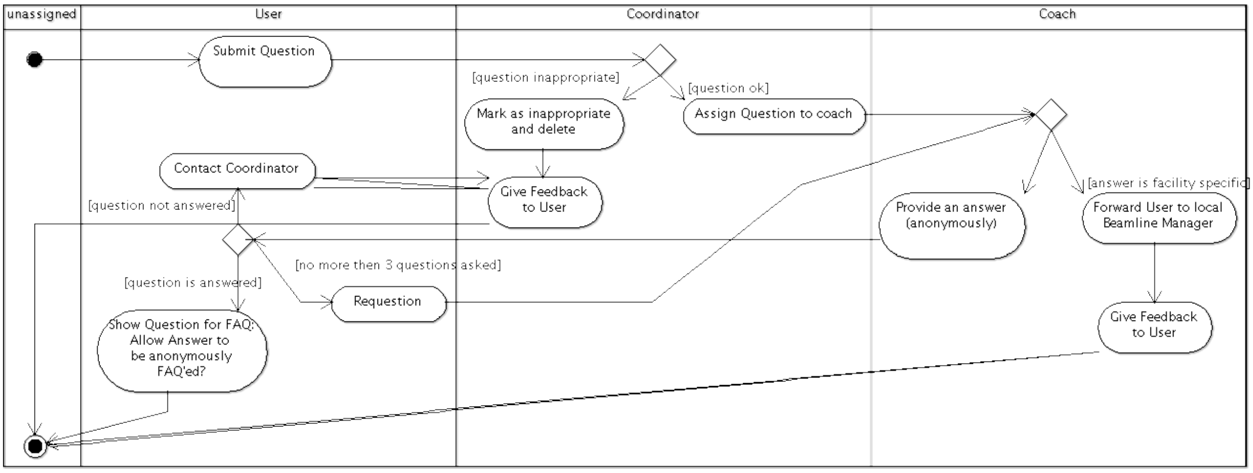

unassigned
User
Coordinator
Coach
Submit Question
[question inappropriate]
[question ok]
Mark as inappropriate and delete
Assign Question to coach
Contact Coordinator
Give Feedback to User
[question not answered]
[answer is facility specific]
Provide an answer (anonymously)
Forward User to local Beamline Manager
[no more then 3 questions asked]
[question is answered]
Requestion
Show Question for FAQ: Allow Answer to be anonymously FAQ'ed?
Give Feedback to User

Illustration 34: Coaching Activities as Workflow

7.3.3. Sequence

This sequence diagram shows the same workflow as in Illustration 3 as an orchestration of system calls.

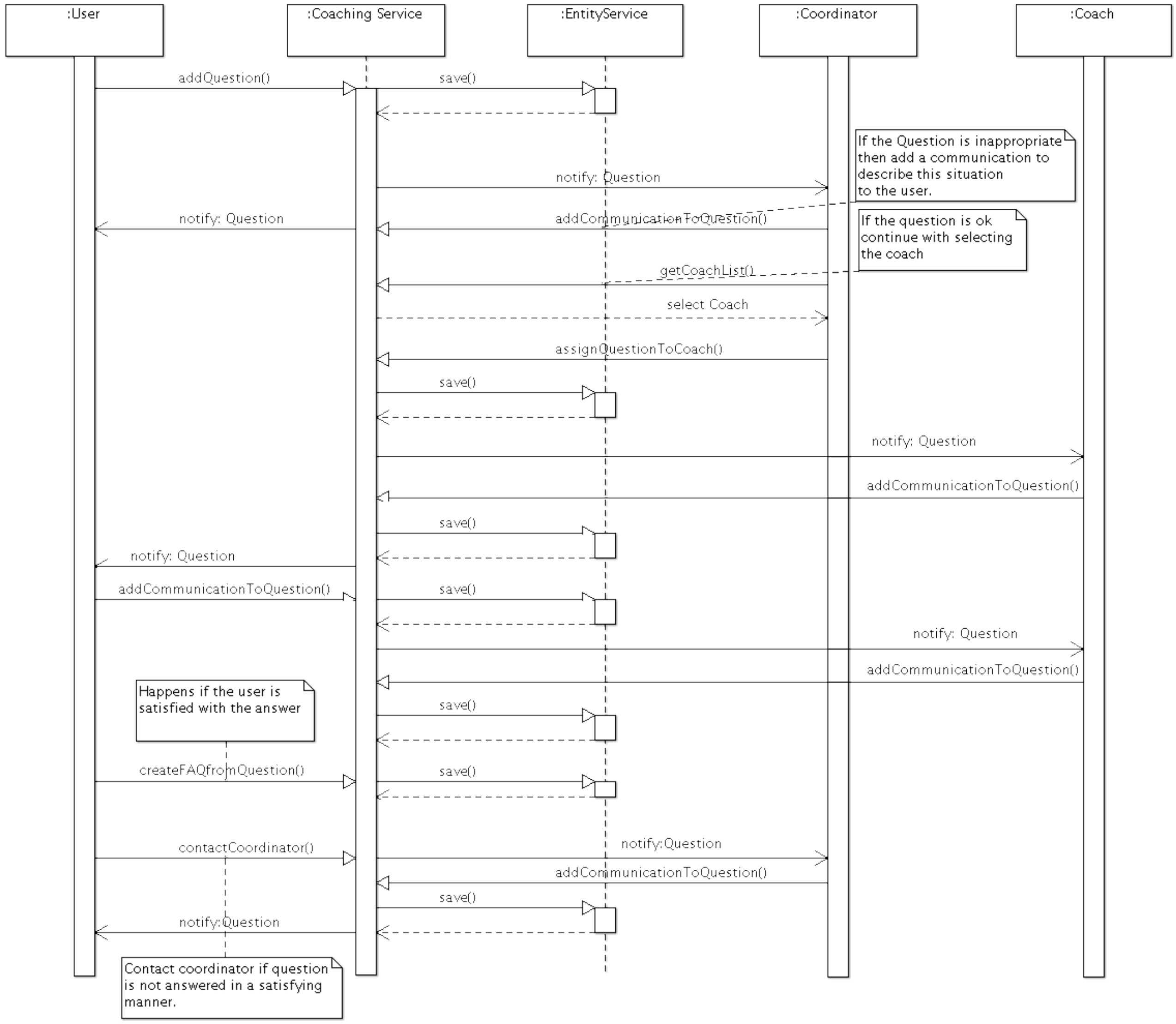

:User
:Coaching Service
:EntityService
:Coordinator
:Coach
addQuestion()
save()
If the Question is inappropriate then add a communication to describe this situation to the user.
notify: Question
notify: Question
addCommunicationToQuestion()
If the question is ok continue with selecting the coach
getCoachList()
select Coach
assignQuestionToCoach()
save()
notify: Question
addCommunicationToQuestion()
save()
notify: Question
addCommunicationToQuestion()
save()
notify: Question
addCommunicationToQuestion()
Happens if the user is satisfied with the answer
save()
createFAQfromQuestion()
save()
contactCoordinator()
notify:Question
addCommunicationToQuestion()
save()
notify:Question
Contact coordinator if question is not answered in a satisfying manner.

Illustration 35: Coaching activities as sequence diagram

7.4. Components

The coaching system is composed of two main components: *FAQ* and *Questioning*. The *FAQ* provides help information and a compilation of successfully answered questions including a search functionality. The *Questioning* provides the functionality for posing a question and the subsequent process for answering the question. The *Answering* needs an authentication- and authorization mechanism and it must use a server side session.

All methods are provided by a coaching service which acts as an façade to underlying services. The class model is injected to the entity service and the database scheme is created automatically using DDL (data definition language).

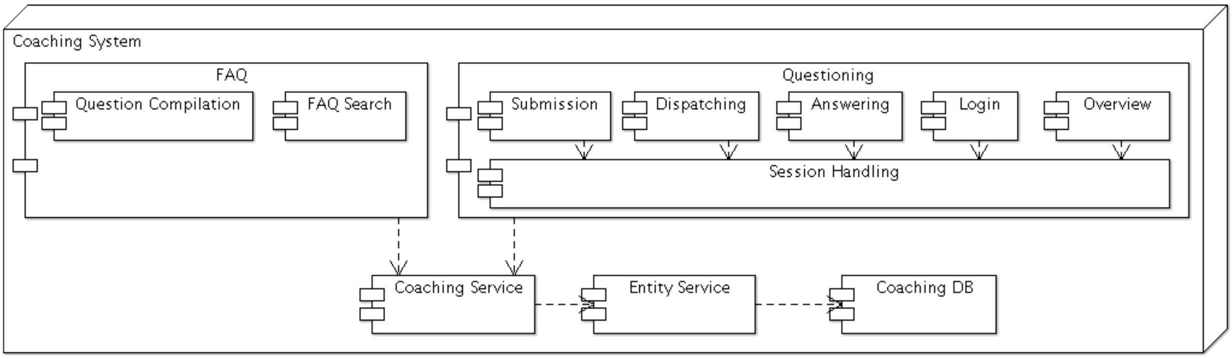


Illustration 36: Components of the coaching system

7.4.1. Coaching Service

Component	**C_C1**
Name	Coaching Service
Identifier	C_C1
Version	0.1
Stereotype	Service
Goal	Deliver the coaching system with functionality and logic
Description	Deliver all functionality needed by the coaching system and abstract from Entity Service.
Input	See Illustration: Class Diagram of Coaching Service
Output	See Illustration: Class Diagram of Coaching Service
Presentation	none
API	RMI, SOAP, REST, IIOP
Actors	User, Coach, Coordinator

Table 41: Component C_C1 Coaching Service

The coaching service provides an API to deal with questions and answers. It could also be accessed from a 3rd party application using standard protocols like SOAP or REST. The coaching service implements the EAA Service interface and is a registered service under the Umbrella.

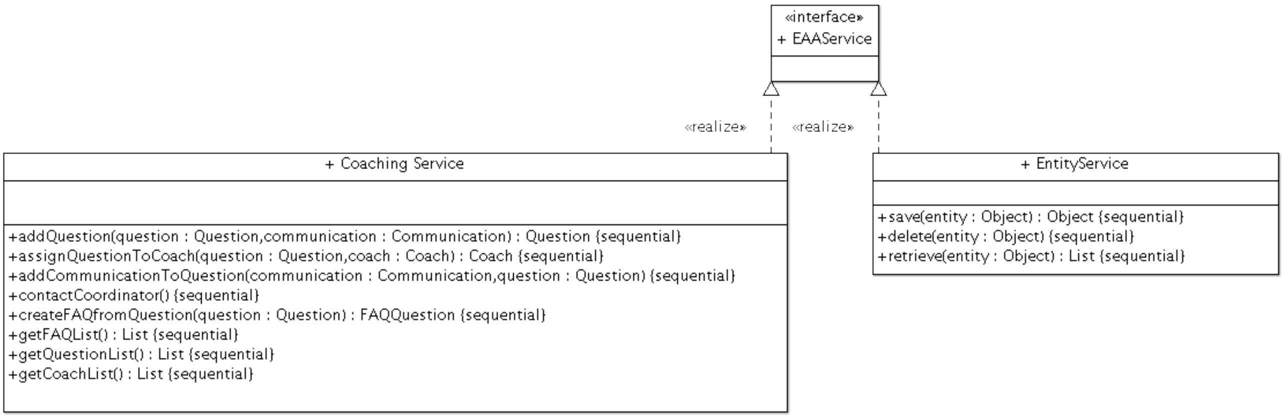

«interface»
+ EAAService
«realize»
«realize»
+ Coaching Service
+addQuestion(question : Question,communication : Communication) : Question {sequential}
+assignQuestionToCoach(question : Question,coach : Coach) : Coach {sequential}
+addCommunicationToQuestion(communication : Communication,question : Question) {sequential}
+contactCoordinator() {sequential}
+createFAQfromQuestion(question : Question) : FAQQuestion {sequential}
+getFAQList() : List {sequential}
+getQuestionList() : List {sequential}
+getCoachList() : List {sequential}
+ EntityService
+save(entity : Object) : Object {sequential}
+delete(entity : Object) {sequential}
+retrieve(entity : Object) : List {sequential}

Illustration 37: Class Diagram of Coaching Service

7.4.2. Entity Service

Component	C_C2
Name	Entity Service
Identifier	C_C2
Version	0.1
Stereotype	Entity
Goal	Deliver core entity system with additional entities used by coaching system
Description	Entity-Model of the coaching system with all entities used. They are mapped to a database by an Object-Relational-Mapping(ORM) tool. The database scheme can be completely generated by this component.
Input	See Illustration: Class Diagram of Coaching Entities
Output	See Illustration: Class Diagram of Coaching Entities
Presentation	none
API	RMI
Actors	Coaching Service

Table 42: Component C_C2 Entity Service

The Illustration “Class Diagram of Coaching Entities” shows the business domain model. It can be injected to an entity service and can be used to create, read, update and delete objects mapped to the database.

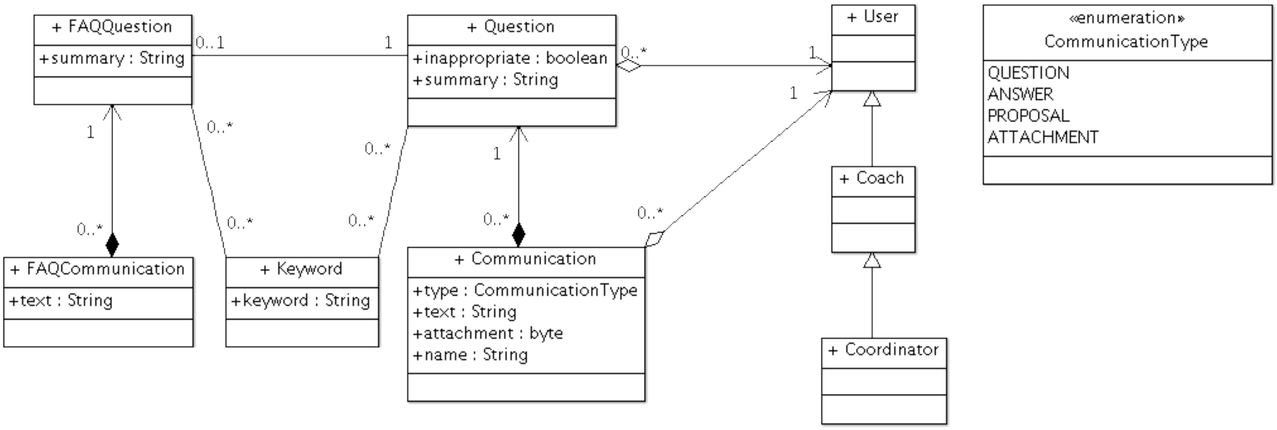


Illustration 38: Class Diagram of Coaching Entities

7.4.3. FAQ Search

Component	C_C3
Name	FAQ Search
Identifier	C_C3
Version	0.1
Stereotype	Web component
Goal	Search FAQ
Description	User can search existing FAQ entries for a text. Coordinator and Coaches are allowed also to find non FAQ'ed entries.
Input	Query:Text
Output	Results:List
Presentation	• Search field • Results page
API	HTTP
Actors	User, Coach

Table 43: Component C_C3 FAQ Search

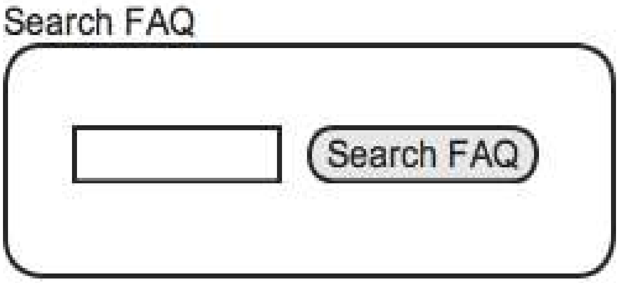
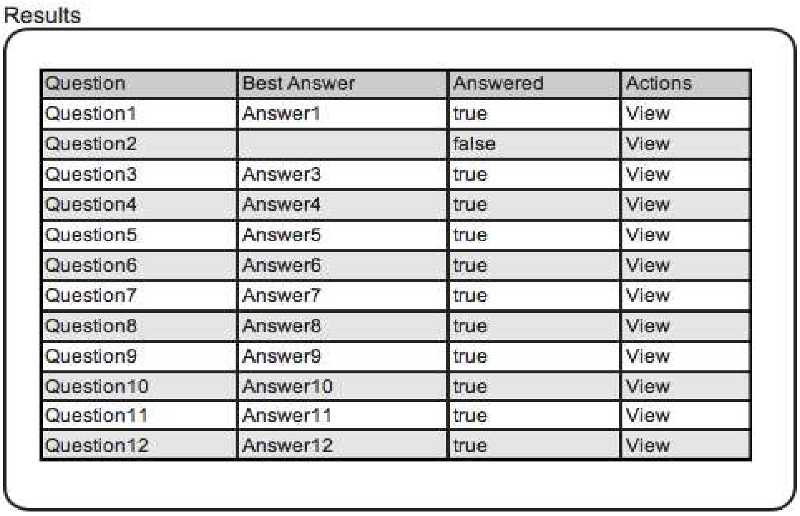


Illustration 39: Component C_C3 Search field



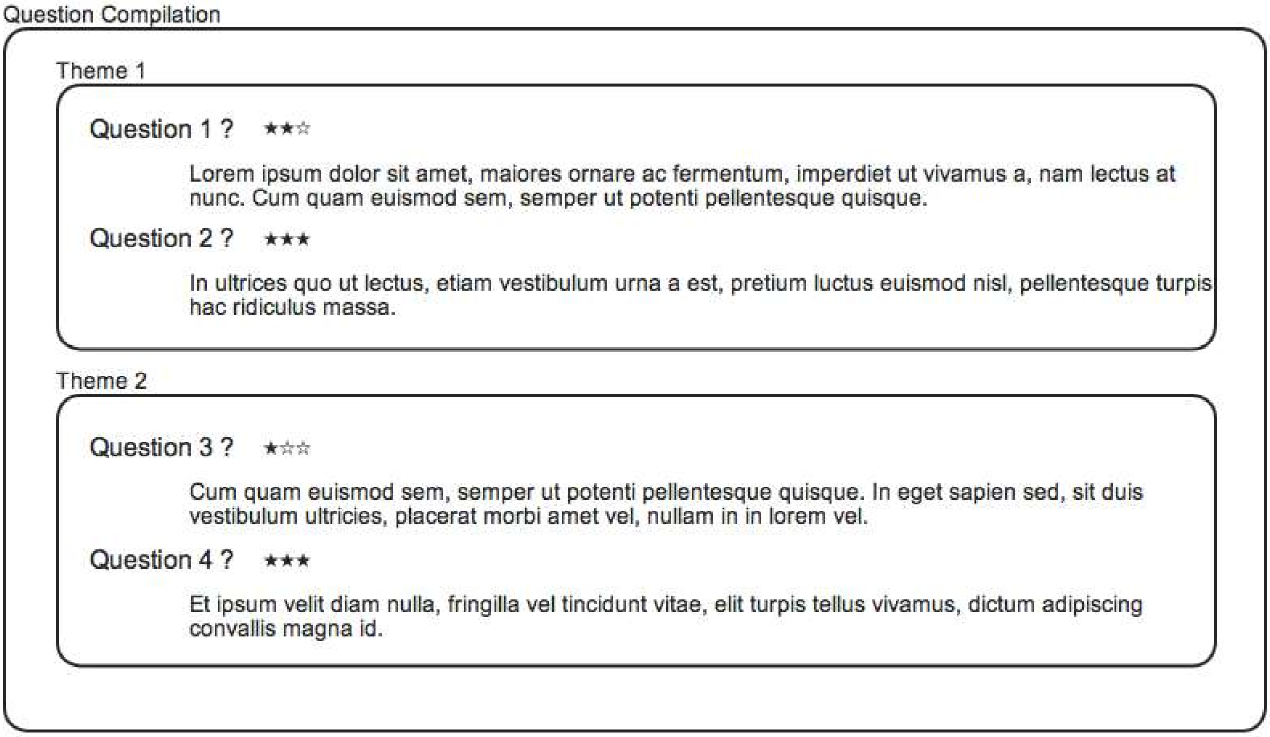
Question	Best Answer	Answered	Actions
Question1	Answer1	true	View
Question2		false	View
Question3	Answer3	true	View
Question4	Answer4	true	View
Question5	Answer5	true	View
Question6	Answer6	true	View
Question7	Answer7	true	View
Question8	Answer8	true	View
Question9	Answer9	true	View
Question10	Answer10	true	View
Question11	Answer11	true	View
Question12	Answer12	true	View

Illustration 40: Component C_C3 Results page

7.4.4. Question Compilation

Component	**C_C4**
Name	Question Compilation
Identifier	C_C4
Version	0.1
Stereotype	Web component
Goal	Compilation of questions for FAQ
Description	This compilation is an aggregation of selected QAs which have been allowed by the user to be used and then refined by coaches.
Input	none
Output	FAQCompilation:List
Presentation	• Question compilation
API	HTTP
Actors	User

Table 44: Component C_C4 Question Compilation



Question Compilation

Theme 1

Question 1 ? ★★☆

Lorem ipsum dolor sit amet, maiores ornare ac fermentum, imperdiet ut vivamus a, nam lectus at nunc. Cum quam euismod sem, semper ut potenti pellentesque quisque.

Question 2 ? ★★★

In ultrices quo ut lectus, etiam vestibulum urna a est, pretium luctus euismod nisl, pellentesque turpis hac ridiculus massa.

Theme 2

Question 3 ? ★☆☆

Cum quam euismod sem, semper ut potenti pellentesque quisque. In eget sapien sed, sit duis vestibulum ultricies, placerat morbi amet vel, nullam in in lorem vel.

Question 4 ? ★★★

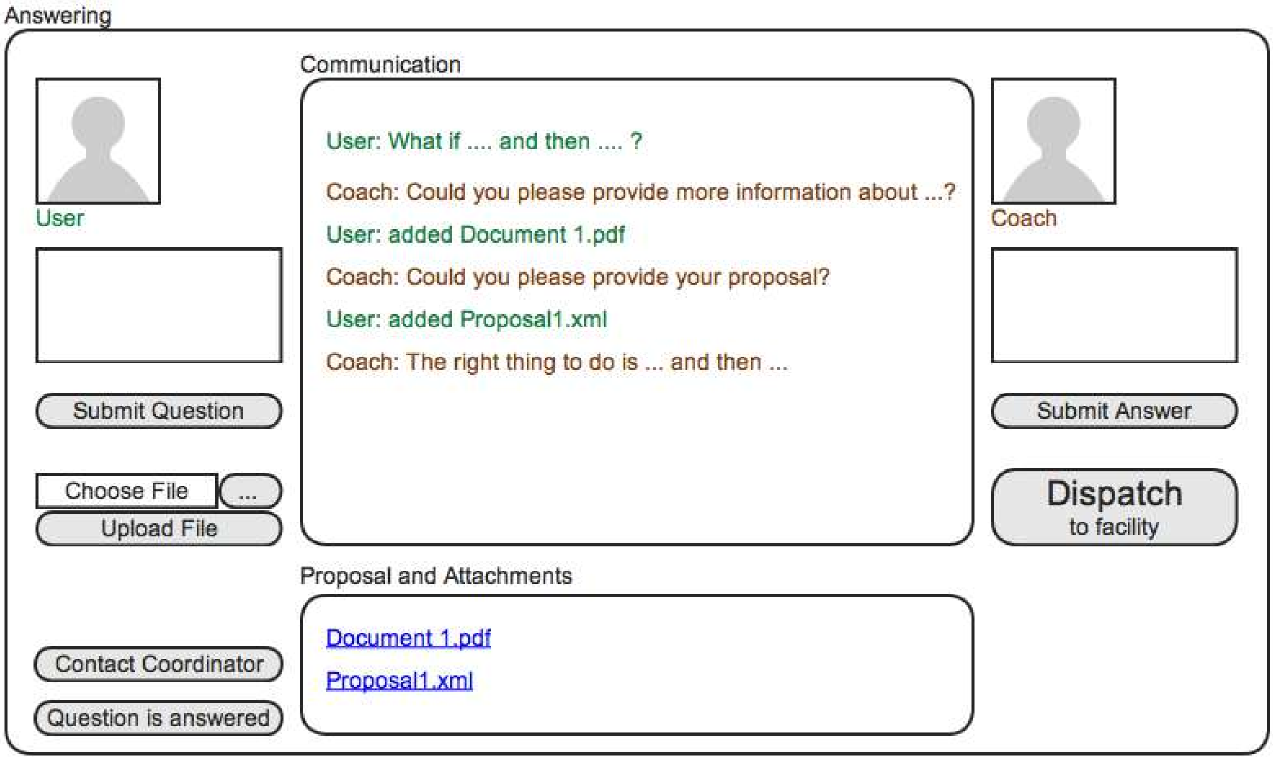
Et ipsum velit diam nulla, fringilla vel tincidunt vitae, elit turpis tellus vivamus, dictum adipiscing convallis magna id.

Illustration 41: Component C_C4 Question Compilation

7.4.5. Answering

Component	C_C5
Name	Answering
Identifier	C_C5
Version	0.1
Stereotype	Web component
Goal	Handle the communication between coach and user.
Description	This web component provides the functionality for handling the communication between users and coaches. The user can view the history of his questioning and add further questions if his iteration count did not reach the limit. He/she may add also attachments (e.g. proposal). The coach can add answers or dispatch the user to a facility. When the iteration count is hit, the user can close this question or contact the coordinator. This screen shows only that information which is related to his/her role (e.g. a user would not see the coach elements and vice versa)
Input	Question:Communication, Answer:Communication
Output	Communication:List
Presentation	• Answering
API	HTTP
Actors	User, Coach

Table 45: Component C_C5 Answering



User: What if .... and then .... ?

Coach: Could you please provide more information about ...?

User: added Document 1.pdf

Coach: Could you please provide your proposal?

User: added Proposal1.xml

Coach: The right thing to do is ... and then ...

Document 1.pdf

Proposal1.xml

Illustration 42: Component C_C5 Answering

7.4.6. Login

Component	C_C6
Name	Login
Identifier	C_C6
Version	0.1
Stereotype	Web component
Goal	Login to the coaching system
Description	In order to identify and authorize an incoming user, a login is performed against the EAA. The session with the coaching system is set up.
Input	Credentials:Text
Output	Session
Presentation	• Login page
API	HTTP
Actors	User, Coach, Coordinator

Table 46: Component C_C6 Answering

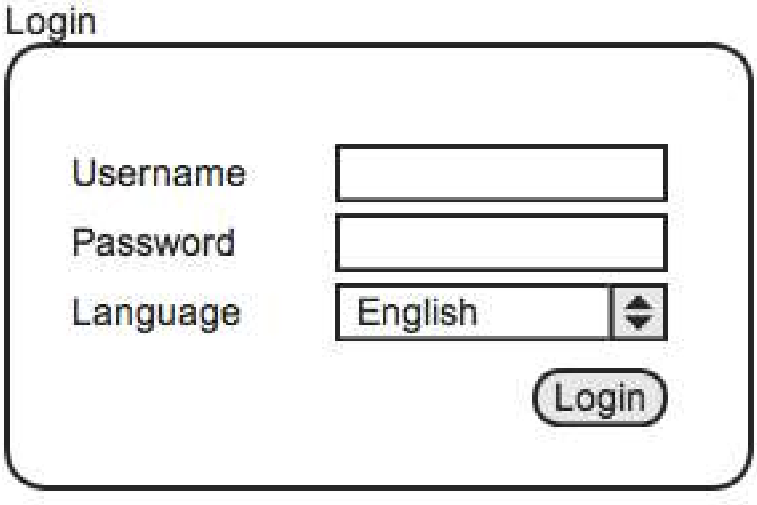
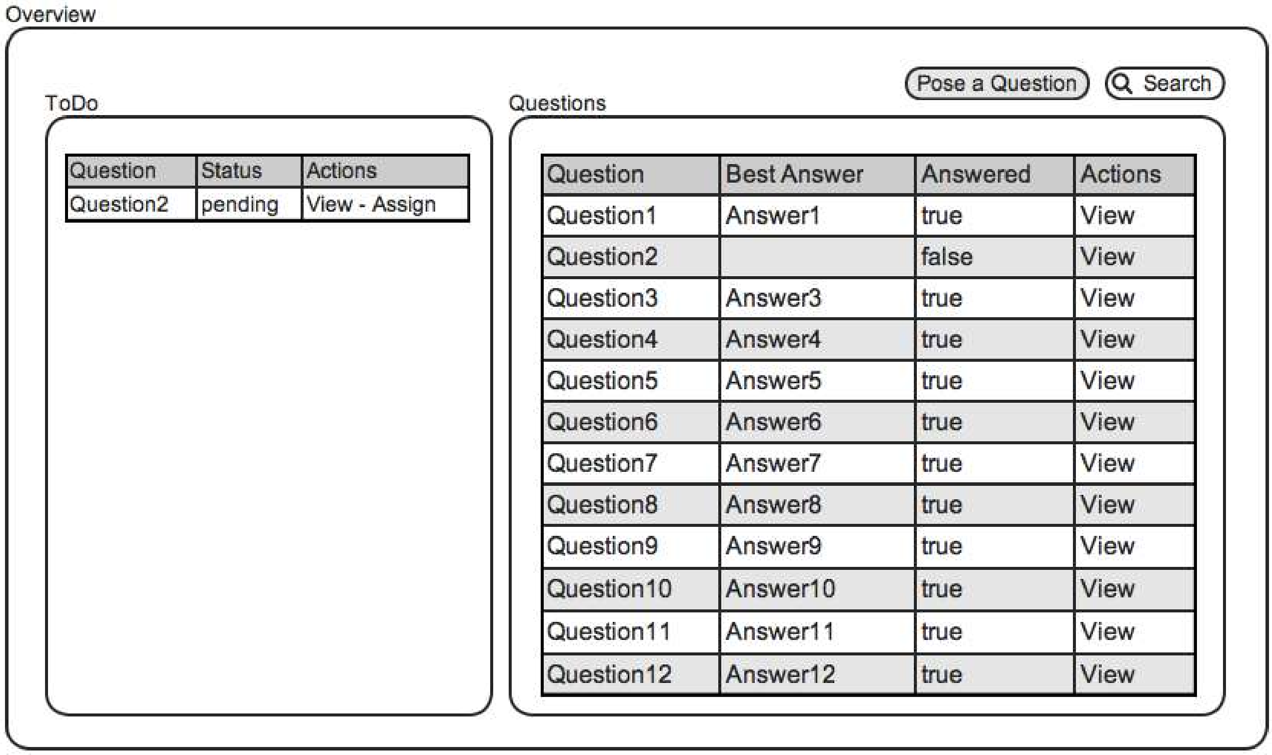


Illustration 43: Component C_C6 Login screen

7.4.7. Overview

Component	C_C7
Name	Overview
Identifier	C_C7
Version	0.1
Stereotype	Web component
Goal	Overview of ToDo's and all posed questions.
Description	This component is the main component of the coaching system. It provides a ToDo list with all open and relevant cases to the user/role combination in the session. It also provides a list of all questions posed yet (if role is user, then only those owned by one self), which can be searched. The “Pose a Question” button allows the submission of a new question.
Input	
Output	Questions:List
Presentation	• Overview
API	HTTP
Actors	User, Coach, Coordinator

Table 47: Component C_C7 Overview



ToDo

Question	Status	Actions
Question2	pending	View - Assign

Questions

Question	Best Answer	Answered	Actions
Question1	Answer1	true	View
Question2		false	View
Question3	Answer3	true	View
Question4	Answer4	true	View
Question5	Answer5	true	View
Question6	Answer6	true	View
Question7	Answer7	true	View
Question8	Answer8	true	View
Question9	Answer9	true	View
Question10	Answer10	true	View
Question11	Answer11	true	View
Question12	Answer12	true	View

Illustration 44: Component C_C7 Overview

7.4.8. Dispatching

Component	C_C8
Name	Dispatching
Identifier	C_C8
Version	0.1
Stereotype	Web component
Goal	Coordinator selects coach to handle question.
Description	The coordinator receives a summary of the question and then selects a coach from a list and assign him to this case. The coach is informed(email) about his/her assignment. A deadline is set up. When this deadline is reached, coordinator and coach are informed.
Input	SelectedCoach:Coach
Output	Coaches:List
Presentation	• Dispatcher
API	HTTP
Actors	Coordinator, Coach

Table 48: Component C_C8 Dispatching

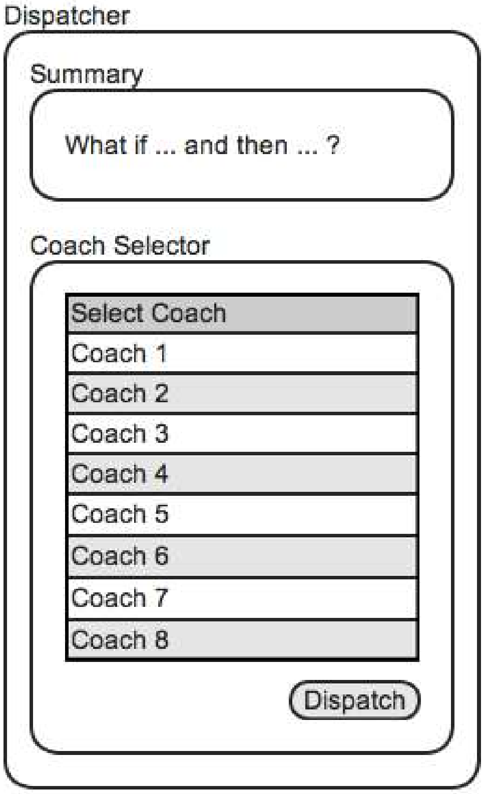


Illustration 45: Component C_C8 Dispatching

7.4.9. Submission

Component	C_C9
Name	Submission
Identifier	C_C9
Version	0.1
Stereotype	Web component
Goal	Submit a new question to coaching system
Description	User can add a question with the help of a question template (to be defined). He/she can add attachments like letter of intent (LoI) or proposal. After submission the question has then to be assigned to a coach by the coordinator.
Input	Question:Question
Output	SubmittedQuestion:Question
Presentation	• Submission
API	HTTP
Actors	User

Table 49: Component C_C9 Submission

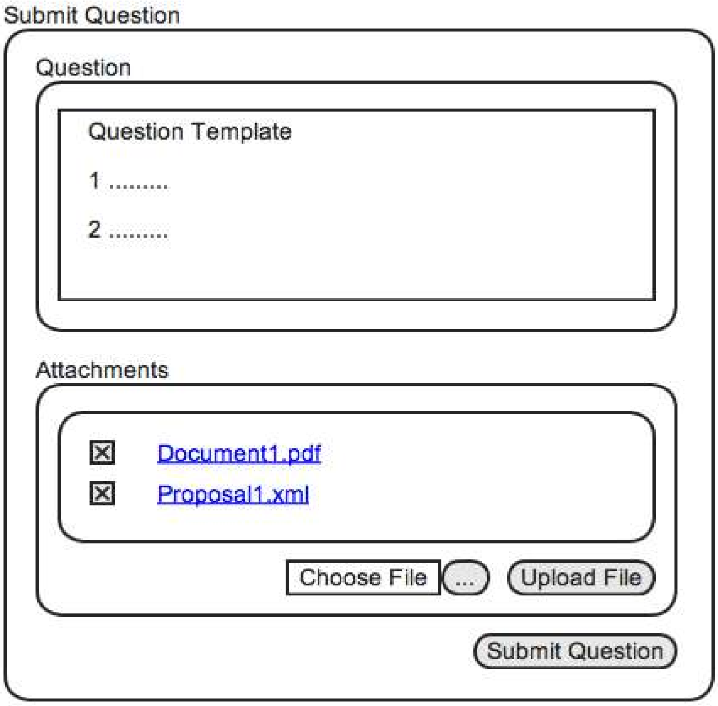


Illustration 46: Component C_C9 Submission

8. U3 Plugin

In order to integrate a single WUO into the U3 system we will provide as much helper functionality as possible. For this to happen in a structured way a plugin should be provided so that update impacts on WUOs are as small as possible.

8.1. Thoughts behind

- The U3 Plugin should help WUOs to easily integrate with U3 by supplying a set of tools.
- A component which checks consistency of the installation is integrated to ensure standardization and safe operations of the WUOs.
- Alternatively a recipe could be provided describing the steps to integrate a WUO to the U3 with the drawback that if the WUO-procedure needs an update it must be done at each WUO. If we utilize a plugin, changes can be made once and distributed.
- An update of the plugin could be announced or automatically installed. Considering the risks of breaking something in the WUO we choose that updates should be announced and then manually installed.

8.2. Use Cases

The U3 Plugin use cases describe the level of integration between WUO and U3 as well as a contract on which the smooth working of the U3 system can be guaranteed.

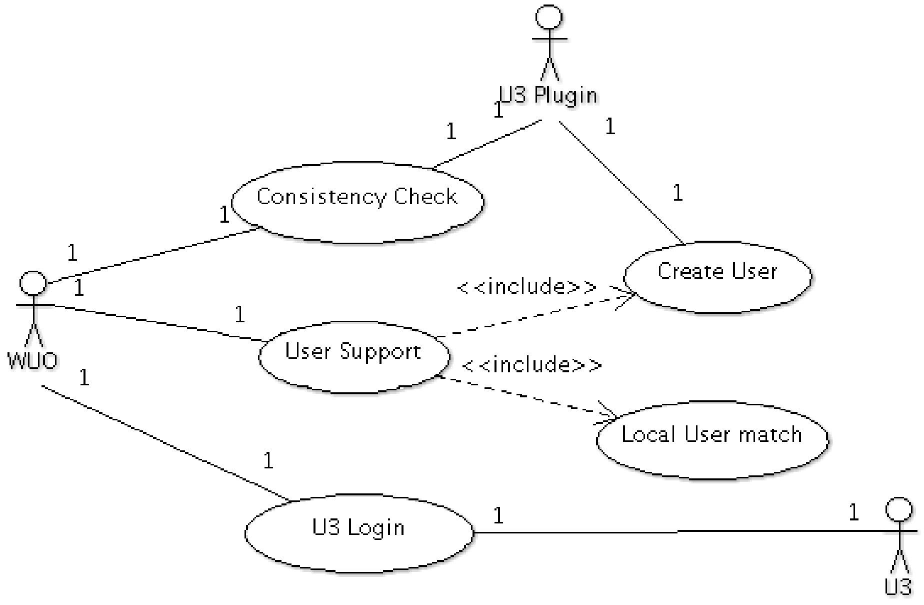


Illustration 47: Use Cases U3 Plugin

Following an overview of the actors involved with the U3 Plugin.

Actors	U3 Plugin
WUO	Web User Office Software
U3	Unified User Umbrella
U3 Plugin	Plugin to integrate WUOs with U3

Table 50: Actors U3 Plugin

8.2.1. Consistency Check

Use Case	UC_P1
Name	Consistency Check
Identifier	UC_P1
Version	0.1
Goal	Verify if the WUO is properly integrated into the U3 System
Summary	The Consistency Check consists of a set of test which indicate the WUOs integration into the U3. It should be able to add missing fields to the WUO-DB.
Actors	U3 Plugin, WUO
Stakeholders	WUO, Community
Preconditions	Plugin must be installed in WUO
Triggers	• Check during installation • Manual triggering
Course of events	1. Consistency check is triggered 2. Check all connections to databases and to U3 3. Verify that all needed fields exist in the database 4. Verify additional “to be defined” conditions 5. Signal successful installation

Table 51: Use Case UC_P1

8.2.2. User Support

Use Case	UC_P2
Name	User Support
Identifier	UC_P2
Version	0.1
Goal	Supply support for user matching and creation at WUO
Summary	Support matching between U3 and WUO users. If U3 user exists and WUO user doesn't then show local WUO login. After successful login, the user entry is loaded with the U3 hash. If the user doesn't exist at the WUO then simply create a user at the WUO. Regarding the decision if a user should be created or a login displayed for user matching there are two possible solutions: • The user must decide if he already has an account • His information must be cross-checked against the data at the WUO-DB and from there forwarded either to a login screen or the user is silently created in the background.

Use Case	UC_P2
Actors	User, U3 Plugin
Stakeholders	User
Preconditions	- Plugin must be installed at WUO User must be logged in at U3
Triggers	Logged in U3 user visits WUO, which doesn't have this U3 user locally registered.
Course of events	1. User enters WUO site 2. User is queried against local user database 3. User is presented with a login screen if candidate is found 4. On successful login users are matched 5. User is created in background if no candidate was found

Table 52: Use Case UC_P2

One goal which the U3 tries to reach is to have a duplicate free user directory. This philosophy should also be applied at the integration level between WUOs and U3 so that the operation of the U3 neither creates duplicate user entries at WUO level.

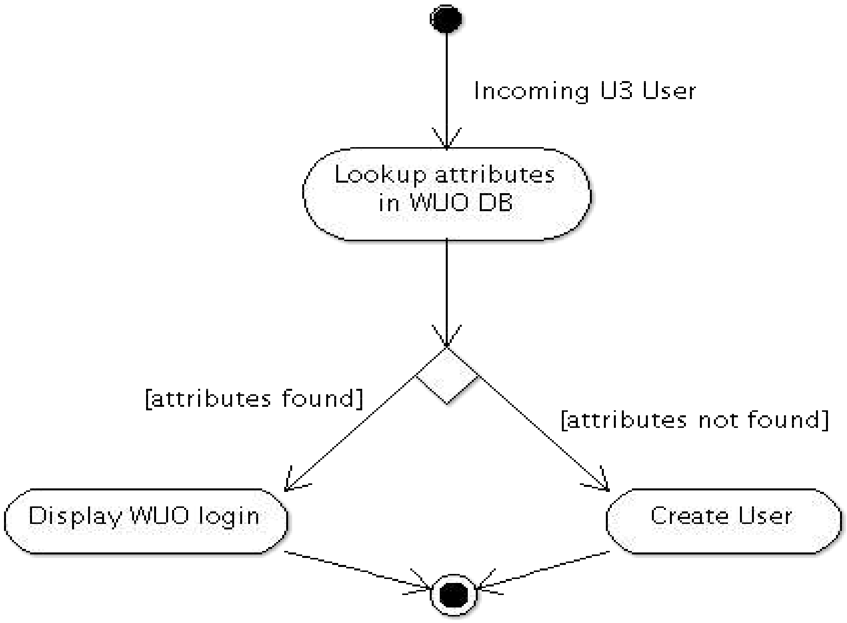


Illustration 48: Activity diagram showing the course of events of the user support use case

8.2.3. Create User

Use Case	UC_P3
Name	Create User
Identifier	UC_P3
Version	0.1
Goal	Create a WUO user with the information provided by U3
Summary	If a non-existing user accesses a WUO and is logged in to the U3 then a user should be created at the WUO, matching the U3 user
Actors	User, WUO
Stakeholders	User, WUO
Preconditions	User must exist and be logged in to the U3
Triggers	Is triggered by User Support
Course of events	1. User accesses WUO 2. User doesn't exist at WUO 3. Silently a user with no authorization is created at the WUO with the supplied information. 4. User is now logged in at the WUO

Table 53: Use Case UC_P3

8.2.4. Local User Match

Use Case	UC_P4
Name	Local User Match
Identifier	UC_P4
Version	0.1
Goal	Match a WUO account to an U3 account
Summary	An existing WUO user is matched to an existing U3 user. For this to happen, a WUO login is displayed and after successful login (verification!) the user database is enriched with the U3 hash which identifies this unique person. After that, no login at the WUO is required anymore when logged in at U3.
Actors	User, WUO U3 Plugin
Stakeholders	User, WUO
Preconditions	• User must be logged in at U3 • User must exist at U3
Triggers	Is triggered by User Support
Course of events	1. User accesses WUO 2. User exists at WUO 3. User is presented with a WUO login 4. If successfully logged in, the user at the database is enriched with the U3 hash 5. User is now matched

Table 54: Use Case UC_P4

8.2.5. U3 Login

Use Case	**UC_P5**
Name	U3 Login
Identifier	UC_P5
Version	0.1
Goal	Provide a method to log in to U3
Summary	To simplify the integration of the plugin into a WUO, the procedure to build up a session with U3 should be provided, that it easily can be integrated into the WUO login procedure.
Actors	WUO, User, U3
Stakeholders	WUO
Preconditions	WUO login procedure must exist
Triggers	User login at WUO
Course of events	1. User enters his credentials at WUO to login 2. WUO transparently dispatches this login to U3 3. On success the user is now logged-in to U3 as well as at the WUO.

Table 55: Use Case UC_P5

8.3. *Specification*

8.3.1. General

For easy integration, the U3 Plugin should be released in all the programming languages in which WUOs are written. It shall be distributed as one file.

8.3.2. Sequence

The login dispatcher should encapsulate the WUOs from calling the U3 login directly. The only thing which needs to be implemented by the WUOs is a method call in the library.

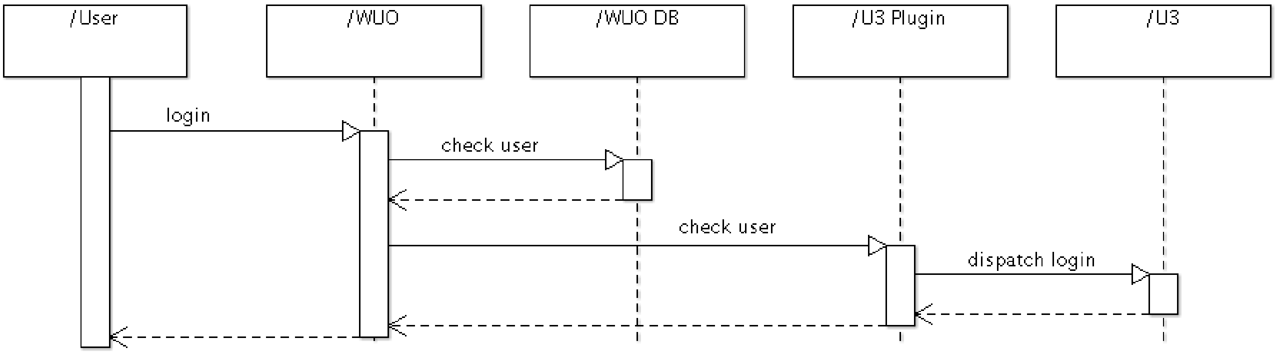


Illustration 49: Login Dispatcher as sequence diagram

8.4. Components

The components of the U3 Plugin are helper utilities for WUOs to integrate with U3.

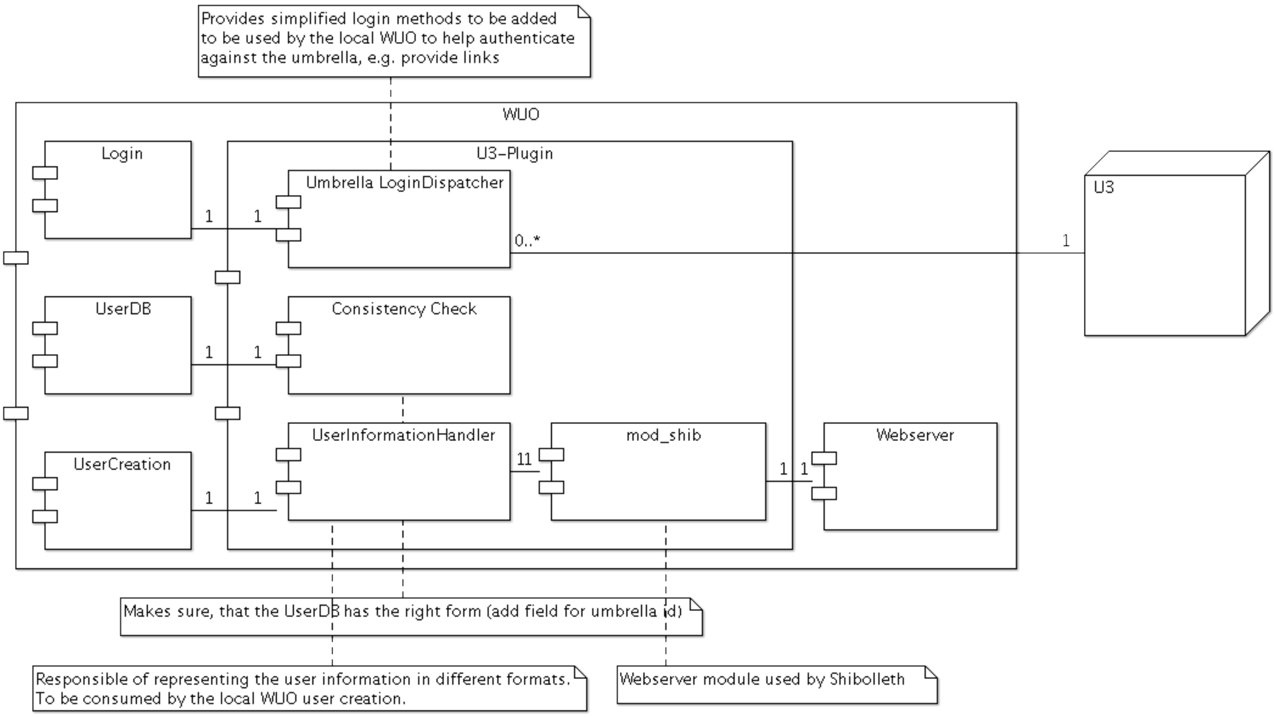

Provides simplified login methods to be added
to be used by the local WUO to help authenticate
against the umbrella, e.g. provide links
WUO
Login
U3-Plugin
Umbrella LoginDispatcher
1
1
0..*
1
U3
UserDB
Consistency Check
1
1
UserInformationHandler
mod_shib
Webserver
11
1
1
UserCreation
1
1
Makes sure, that the UserDB has the right form (add field for umbrella id)
Responsible of representing the user information in different formats.
To be consumed by the local WUO user creation.
Webserver module used by Shibolleth

Illustration 50: Components U3 Plugin

Different functionalities are collected in this library and should be held in different packages in the plugin. A consistency check is provided to ensure proper installation.

8.4.1. Umbrella Login Dispatcher

Component	C_P1
Name	Umbrella Login Dispatcher
Identifier	C_P1
Version	0.1
Stereotype	Authenticator
Goal	Dispatch WUO login to U3
Description	Provide a method for a WUO to chain its authentication with a U3 login. It should build up the conversation with U3 and return a U3 session on success.
Input	User credentials
Output	U3 Session
Presentation	none
API	Core
Actors	WUO, U3

Table 56: Component C_P1 Umbrella Login Dispatcher

8.4.2. Consistency Check

Component	C_P2
Name	Consistency Check
Identifier	C_P2
Version	0.1
Stereotype	Script
Goal	Ensure consistency of plugin installation
Description	In order to function properly with U3, WUOs need a certain level of integration. To ensure that all requirements of this level are properly integrated, a consistency check is performed. This level might change with future releases of U3.
Input	Configuration file
Output	Validity: Boolean
Presentation	none
API	Core
Actors	WUO

Table 57: Component C_P2 DB-Manipulator

We will need a configuration file where properties of the local installation can be entered. These properties then are used to do checks against following:

- database correctness
- inquiries on the local file system
- network installation correctness

8.4.3. User Information Handler

Component	**C_P3**
Name	User Information Handler
Identifier	C_P3
Version	0.1
Stereotype	Class
Goal	Handle incoming user information for matching and creation of users.
Description	Helper class to handle local creation and matching of U3 users with WUO users. WUO login screen is used for user verification
Input	User attributes from mod_shib.
Output	Matched users.
Presentation	none
API	core
Actors	WUO

Table 58: Component C_P3 User Information Handler

8.4.4. mod_shib

Component	C_P4
Name	mod_shib
Identifier	C_P4
Version	0.1
Stereotype	Web server module
Goal	Read and verify incoming session against U3
Description	This module is part of the shibboleth distribution. It is installed at a web server and reads incoming attributes and verifies them against U3.
Input	HTTP-Request
Output	Verified session
Presentation	none
API	HTTP
Actors	U3, WUO

Table 59: Component C_P4 mod_shib

9. Implementation proposal

The platform and language to use is not defined yet. We would like to collect pros and cons of different languages to make a proper decision.

Feature	PHP	Java	.Net
1) Compiled Code	3rd party tool	yes	yes
2) Object Oriented	somewhat	yes	yes
3) Open Source	yes	yes	no
4) Open Stack (no vendor lock-in)	yes	yes	no
5) Development Tools	yes	yes	yes
6) Decent ORM Tools (Object Relational Mapping)	somewhat	yes	yes
7) Support major RDBMS (Oracle, MSSQL, MySQL)	yes	yes	yes
8) Popularity (registered projects on sourceforge.net)	28,548	43,513	12,488
9) Stability of Language: formal specification by committee?	no (done by The PHP Group)	Language and Platform: yes (jcp.org)	Language: yes (ECMA, ISO) Platform: no (MSFT)
10) Many different library providers	yes	yes	somewhat
11) Software vendors using it	IBM	IBM, Oracle (SUN), SAP, Google	Microsoft, SAP

Table 60: Comparison of Programming Languages

9.1. Arguments

- 1) Compiled Code: the code is translated to machine language before it is executed. Influences performance of application. This is in favor of Java/.Net but there is a 3rd party tool also for PHP.
- 2) Object Oriented: Both Java and .Net are strictly Object Oriented. PHP is procedural with an OO extension.
- 6) Object Relational Mapping: is a technology concept to map objects to a database. There is a PHP extension for ORM called doctrine which is catching up in functionality.

From that it follows that there is a preference of Java over the other options unless there are strong additional arguments. Feedback is highly welcome!

10. What will the future bring?

The platform described in this document delivers a stable environment for adding further developments on top. Some possible scenarios are described here:

10.1. Remote Access

With increasing amounts of data delivered by modern detectors, it is almost impossible for users to store everything on a personal hard disc for transportation home. Therefore, a possible future scenario is to store data from experiments centrally and allow users to login remotely to the facility where the experiment has been performed and to continue data processing on site. Another future option will be to remotely access by allowing remote users to login at a facility to monitor and supervise an experiment or even change parameters remotely. This could also be helpful for distributed teams to decrease travel time or to allow senior scientists to participate in an experiment.

10.2. Indico

The CERN Document Server Software Consortium developed a web application to schedule and organize events, from simple lectures to complex meetings, workshops and conferences with sessions and contributions. The tool also includes an advanced user delegation mechanism, to allow paper reviewing, archival of conference information and electronic proceedings. The Indico software was originally developed in the framework of the EU InDiCo project. Nowadays, Indico is a free software licensed under terms of GNU General Public License (GPL). Indico is currently in production at CERN at http://indico.cern.ch.

Indico might also be a candidate to be implemented as a service provider into U3 to provide scheduling and event organization to the community.

10.3. User forum

The members of WP2 agreed that EuroFEL should think of setting up a web forum, although at the current state the real interest is not clear. One should, therefore, start on a low level and increase the effort in case of strong interest. This should be done together with WP5. Also, a moderator would be needed.

10.4. Beamline description tool (analogous to Elisa)

The IT investment is moderate but in order to have a real living tool the data would require continuing curation. Therefore, for the moment also this issue has been assigned 2nd priority.

11. Glossary

1	Activity Diagram	Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.
2	Actor	An actor specifies a role played by a person or thing when interacting with the system.
3	API	An application programming interface (API) is an interface implemented by a software program which enables it to interact with other software. It facilitates interaction between different software programs similar to the way the user interface facilitates interaction between humans and computers
4	Application Layer	The Application Layer describes software components which directly interact with a user.
5	Authentication	Who am I; the goal is to identify a person uniquely Europe-wide. See also EAA
6	Authorization	Which roles (i.e. rights) do I have (e.g. facility access, computer resources), what is my function.
7	Central Affiliation Database	The proposed database, which contains the address information (e.g. postal address) in a standardized format.
8	CI/CD	A corporate identity (CI) is the "persona" of a corporation which is designed to accord with and facilitate the attainment of business objectives. It is usually visibly manifested by way of branding and the use of trademarks. A corporate design (CD) is the official graphical design of the logo and name of a company or institution used on letterheads, envelopes, forms, folders, brochures, etc. The house style is created in such a way that all the elements are arranged in a distinguished design and pattern.
9	Class Diagram	A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.
10	Coaching	Structured support of novice facility users by peer coaches, managed by the EUU / UUU.
11	Credential	A Credential is the amount of information needed by an authentication system to successfully identify a user.
12	Database Layer	The Database Layer provides access to databases in a uniform

		way.
13	DDL	A Data Definition Language or Data Description Language (DDL) is a computer language for defining data structures. It was used to refer to a subset of SQL, but is now used in a generic sense to refer to any formal language for describing data or information structures, like XML schemas.
14	Deployment Diagram	A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes.
15	Directory	Generally, a directory, as used in computing and telephony, refers to a repository or database of information which is heavily optimized for reading, under the assumption that data updates are very rare compared to data reads. Commonly, a directory supports search and browsing in addition to simple lookups.
16	EAA	European Authentication and Authorization, service, which uniquely identifies a user within the European Neutron and Photon user community. Authentication is via a central portal (Umbrella), authorization is performed by the local WUOs. In order to minimize the administration overhead authentication is realized as multi-level authentication.
17	Entity	An entity may be defined as a thing which is recognized as being capable of an independent existence and which can be uniquely identified. An entity is an abstraction from the complexities of some domain. When we speak of an entity we normally speak of some aspect of the real world which can be distinguished from other aspects of the real world.
18	ESRFUP	ESRF upgrade program, one of the projects within the FP7 roadmap program.
19	EUU	EuroFEL User Umbrella, prototype for the UUU for the members of the EuroFEL consortium; project within WP2 of the IRUVX-PP EU Roadmap program.
20	Façade (Façade Pattern)	The façade pattern is a software engineering design pattern commonly used with Object-oriented programming.A façade is an object that provides a simplified interface to a larger body of code, such as a class library.
21	FAQ	Frequently asked questions, or FAQs are listed questions and answers, all supposed to be frequently asked in some context, and pertaining to a particular topic.
22	Hard Authentication	In-person identification of a user, e.g. via a legal document at a local user office.
23	Hash	A hash function is any well-defined procedure or mathematical function that converts a large, possibly variable-sized amount of data into a small datum. Hash functions are mostly used to speed up table lookup or data comparison tasks—such as

		finding items in a database, detecting duplicated or similar records in a large file, finding similar stretches in DNA sequences, and so on.
24	HTTP	The Hypertext Transfer Protocol (HTTP) is a networking protocol for distributed, collaborative, hypermedia information systems.[1] HTTP is the foundation of data communication for the World Wide Web.
25	HTTP POST	HTTP POST is one of many request methods supported by the HTTP protocol used by the web. The POST request method is used when the client needs to send data to the server as part of the request, such as when uploading a file or submitting a completed form.
26	Identity Provider	An Identity Provider is a Shibboleth server that authenticates users and conveys their attributes to requesting resources. In other terms it provides the digital identities of its users to other servers in the federation.
27	IIOP	IIOP (Internet Inter-Orb Protocol) is a protocol by which object request brokers (ORBs) communicate over TCP/IP.
28	IRUVX-PP	EuroFEL Consortium Preparatory Phase, one of the projects of the FP7 roadmap program.
29	JavaScript	JavaScript is an implementation of the ECMAScript language standard. JavaScript is primarily used in the form of client-side JavaScript, implemented as part of a web browser in order to provide enhanced user interfaces and dynamic websites.
30	Layer (Software)	A layer (or abstraction layer) is a way of hiding the implementation details of a particular set of functionality. The simplification provided by a good layer allows for easy reuse by distilling a useful concept or metaphor so that situations where it may be accurately applied can be quickly recognized.
31	LDAP	The Lightweight Directory Access Protocol (LDAP) is an application protocol for querying and modifying data of directory services. It is specified in a series of Internet Engineering Task Force (IETF) Requests for comments (RFC) as detailed in RFC4510.
32	mod_shib	Web server module handling Shibboleth requests on Service Provider side.
33	Multi-Level Authentication	In order to minimize the administration overhead for users and management, authentication is realized as multi-level authentication. Soft authentication is sufficient for services like registration for news, conference services, hard registration e.g. for access to facility sites, beamline components. The details for a specific site are defined by the local management.
34	ORM	Object-relational mapping (ORM, O/RM, and O/R mapping) is a

		programming technique for converting data between incompatible type systems (relational model) in object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language.
35	PaaS	Platform as a service (PaaS) describes the delivery of a computing platform and solution stack as a service.
36	Public Key Infrastructure (PKI)	Public Key Infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.
37	Push-Only Operation	The EAA system will be set up in such a way, that the central portal, triggered by a user, can push information to a local WUO. Pulling of central information by a local WUO is not allowed.
38	QA	Question answering (QA) refers to the task of answering questions in natural language.
39	RDBMS	A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd. Most popular commercial and open source databases currently in use are based on the relational database model.
40	Repository	A repository is a library in which collections are stored in digital formats and are accessed by computers. The content may be stored locally, or accessed remotely via computer networks. A repository is a type of information retrieval system.
41	REST	Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. The term Representational State Transfer was introduced and defined in 2000 by Roy Fielding in his doctoral dissertation. Fielding is one of the principal authors of the Hypertext Transfer Protocol (HTTP) specification versions 1.0 and 1.1.
42	RMI	The Remote Method Invocation (RMI) is an object-oriented inter-process communication that allows a computer program to cause a subroutine or procedure to execute in another address space (commonly on another computer on a shared network) without the programmer explicitly coding the details for this remote interaction.
43	Runtime Environment	The runtime environment is a collection of software services available. These services may be provided by the operating system, or by a run-time system, such as a virtual machine or a collection of program libraries.
44	SAML2	Security Assertion Markup Language (SAML) is an XML-based

		standard for exchanging authentication and authorization data between security domains, that is, between an identity provider (a producer of assertions) and a service provider (a consumer of assertions). SAML is a product of the OASIS Security Services Technical Committee.
45	Sequence Diagram	A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order.
46	Service	A service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description.
47	Service Layer	The Service Layer contains services exposed to applications. It's main purpose is to provide uniform access to algorithms and data
48	Service Provider	A Service Provider is a service inside the Shibboleth federation. Authenticated users can consume them.
49	Session	A session is a semi-permanent interactive information interchange, also known as a dialogue, a conversation or a meeting, between two or more communicating devices, or between a computer and user. A session is set up or established at a certain point in time, and torn down at a later point in time.
50	Shibboleth	Federated open-software authentication system, basis of the handshake between the central portal and the local WUOs.
51	Single Sign On	A user, who has once been authenticated by logging on at the central portal, has access to the services he/she is authorized to without need for additional authentications.
52	SOAP	SOAP, originally defined as Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks.
53	Soft Authentication	Standard forum-type identification, e.g. via email handshake.
54	Soft User	A Soft User is a user with a minimum of authorization.
55	SQL	SQL, often referred to as Structured Query Language, is a database computer language designed for managing data in relational database management systems (RDBMS), and originally based upon relational algebra
56	Swimlane	A swimlane is a visual element used in process flow diagrams that depict what or who is working on a particular subset of a process.
57	Umbrella	Concept of combining local IT funtionalities to a central common

	concept	funtionality by keeping the local functionalities as much as possible and by adding novel central functionalities.
58	Use Case	A use case is a description of a system’s behavior as it responds to a request that originates from outside of that system. In other words, a use case describes "who" can do "what" with the system in question.
59	UUU, Triple-U	Unified User Umbrella, final tool for EU-wide services for the neutron and photon community, including common portal for proposal handling, coaching, authentication services.
60	Web application framework	A web application framework is a software framework that is designed to support the development of dynamic websites, Web applications and Web services. The framework aims to alleviate the overhead associated with common activities performed in Web development.
61	WUO	Web-based User Office, the application that supports a local user office in managing user-related issues (e.g. proposal handling, on-site user issues).
62	XML	Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures.
63	XSD (XML Schema)	XML Schema (XSD) is one of several XML schema languages. XSD can be used to express a set of rules to which an XML document must conform in order to be considered 'valid' according to that schema. However, unlike most other schema languages, XSD was also designed with the intent that determination of a document's validity would produce a collection of information adhering to specific data types.
64	XSLT	XSLT (Extensible Stylesheet Language Transformations) is a declarative, XML-based language used for the transformation of XML documents into other XML documents. The original document is not changed; rather, a new document is created based on the content of an existing one.

12. Appendix

12.1. How the use cases are described

Attribute	Description
Name	A descriptive name provides a unique identifier for this use case. Can be changed during development.
Identifier	An immutable number. The effect of using immutable numbers is to isolate external references to the use case when names are changed, and to ensure distinct names across many similar use cases
Version	A version section is needed to inform the reader of the stage a use case has reached
Goal	Without a goal a use case is useless. There is no need for a use case when there is no need for any actor to achieve a goal. A goal briefly describes what the user intends to achieve with this use case.
Summary	A summary is used to capture the essence of a use case. It provides a quick overview, which is intended to save the reader from having to read the full contents of a use case to understand what the use case is about.
Actors	An actor is someone or something outside the system that either acts on the system – a primary actor – or is acted on by the system – a secondary actor. An actor may be a person, a device, another system or sub-system, or time.
Stakeholders	A stakeholder is an individual or department that is affected by the outcome of the use case
Preconditions	A preconditions section defines all the conditions that must be true for the trigger to meaningfully cause the initiation of the use case
Triggers	A 'triggers' section describes the event that causes the use case to be initiated.
Course of events	At a minimum, each use case should convey a primary scenario, or typical course of events, also called "basic flow", "normal flow," "happy flow" and "Happy path".

Table 61: How Use Cases are described

12.2. How the components are described

Attribute	Description
Name	A descriptive name provides a unique identifier for this component. Can be changed during development.
Identifier	An immutable number. The effect of using immutable numbers is to isolate external references to the component when names are changed, and to ensure distinct names across many similar use cases
Version	A version section is needed to inform the reader of the stage a component has reached
Stereotype	A stereotype describes the kind of specialization a component has.
Goal	A goal briefly describes what this component tries to achieve
Description	Provide a quick overview of the component
Input	Describes the components input
Output	Describes the output of the component.
Presentation	Describes how and if the component is rendered.
API	Shows the protocols over which this component is accessible.
Actors	An actor is someone or something outside the system that either acts on the system – a primary actor – or is acted on by the system – a secondary actor. An actor may be a person, a device, another system or sub-system, or time.

Table 62: How Components are described

12.3. Illustrations

Illustration Index

12.4. Tables

Index of Tables