UMBRELLA FOR SERVICE INTEGRATORS
DESCRIPTION

Foreword

This document describes the integration steps necessary to integrate a WUO into the Umbrella system. It is split into four parts – one describing how to integrate the existing WUO into the Single Sign-On and Account Linking processes, another describing the Address Updater functionality, another one describing how to upgrade an existing account WUO account to an Umbrella account and last but not least guidelines for the graphic representation.

If not differently noted, following fonts are used:

Arial	-	used for text
`Courier`	-	`used for source code`
`Courier(red)`	-	`used to mark changes to source code`

Introduction

The focus of this paper is to provide a recipe to integrate existing WUOs into the Umbrella system.

A key feature of the Umbrella system is to uniquely identify a user EU- and potentially world-wide. At the moment this is not possible, because all facilities have an own user database with no link between accounts (Figure 1).

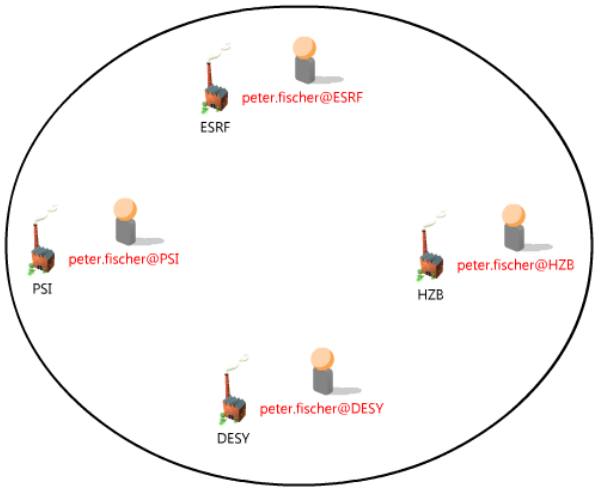


Figure 1: Situation without Umbrella

The goal is to identify the users just once and use this identity on subsequent visits to different facilities (as shown in Figure 2) and still allow existing users who don't participate at the Umbrella to use the WUOs (hybrid aspect).

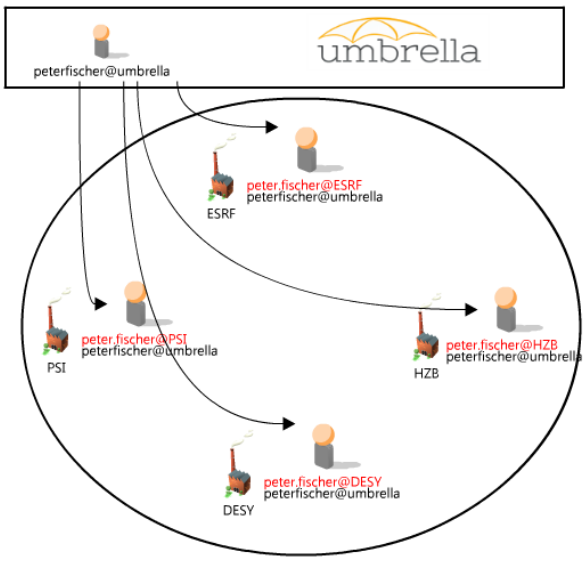


Figure 2: Situation with Umbrella

To virtually “travel” from facility to facility a “Digital ID Card” is issued and used to identify the user at a facility. The released attributes in this “Digital ID Card” look as follows:

Attribute Name	Attribute Description
uid	The users uid in plain text. Can be used to display the username
EAAHash	A UUID generated by Java’s `UUID.randomUUID()`. This unique and persistent identifier is used for matching an existing Umbrella account with an existing WUO account.
EAAKey	A UUID generated by Java’s `UUID.randomUUID()`. This key is used to verify if a user is registered at a specific WUO by a challenge response handshake.

Single Sign-On and Account linking

Concept

To be able to participate at the Umbrella the WUOs must install a SAML2 Service Provider software, e.g. mod_shib2 (Apache) or SimpleSAMLphp (PHP), to be able to receive SAML2 attributes. The metadata of this Service Provider must then be installed in the Umbrella IdP.

Since the page should be accessible when no EAA-Session exists, the SAML2isPassive value must be set.

To enable the hybrid characteristics of the authentication system both authenticating at the WUO and the Umbrella must be possible. This is accomplished through a multi-phase check (see Figure 3) at the WUO itself:

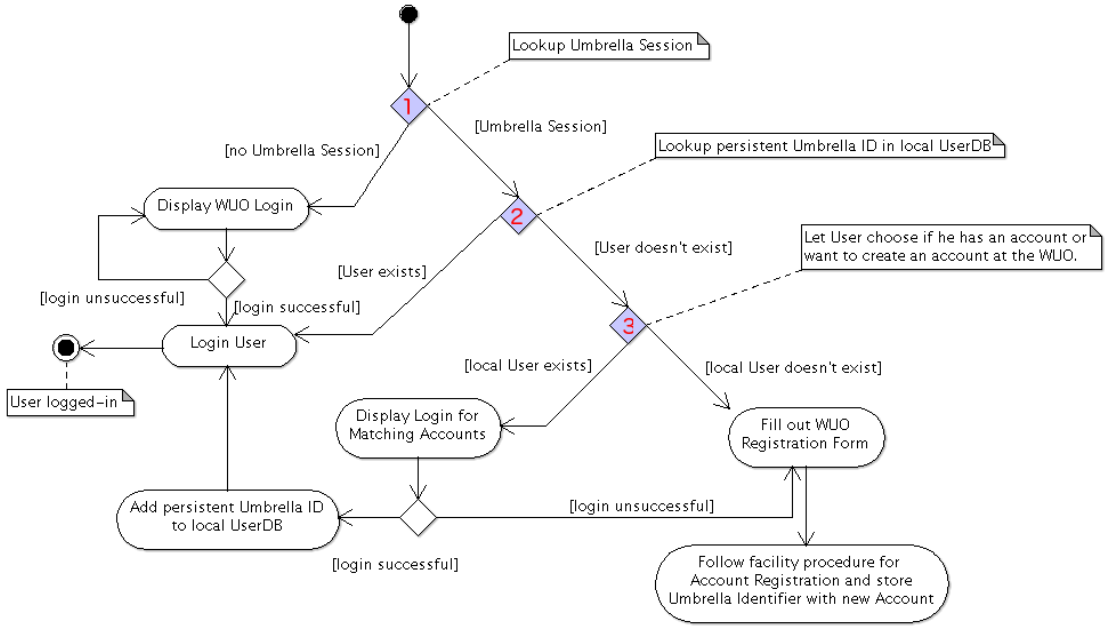


Figure 3: Web SSO User Creation Pattern

It is presumed that following procedures are already installed at the WUOs as this integration builds upon them:

- Login Form
- Registration Form

The Login Form is shown to incoming Umbrella-Users who have a WUO-Account to let them match the two accounts.

The Create Account Form is shown to incoming Umbrella-Users who don't have a WUO-Account to let them create a new account according to the guidelines of the WUO.

Consider the red numbers in Figure 3:

1. Umbrella-Session Check: Check if there is an existing Umbrella -Session active. If there is no valid Umbrella -Session, continue with the normal application code. Else forward the user to User Check.
2. User Check: Query the database for the incoming Umbrella-Hash. If there is a matching user, log him in. Else forward the user to User Matching.
3. User Matching: Let the user choose to match his Umbrella-Account with an existing WUO-Account or let him create a new account.

Examples:

Umbrella-Session Check

To find out if a user has a session, we need to read HTTP server headers for the attribute EAAHash. Following a few examples in different languages:

Language	Construct
Java	`httpServletRequest.getHeader("EAAHash");`
PHP	`$_SERVER["EAAHash"];`

User Check

The User Check consists basically of an extension to the SQL SELECT statement which besides querying for username and password also queries for the incoming EAAHash.

Following code snippet (red and bold) can be used to enhance this generic user query:

```
SELECT
      USERNAME
FROM
      USERS
WHERE
      (USERNAME='user' AND PASSWORD='changeit')
OR
      (EAAHASH='f5bba3c6-6240-4ccf-8048-13dbb3405192')
```

User Matching

User Matching is necessary if there is an incoming Umbrella-Session but no WUO user registered with it. It's important to use the existing “Login” and “Create User” procedures already installed at the facilities, so that no new procedures must be installed and approved.

There is a chance that the user already has an existing account at the WUO and that the user wants to bind those accounts together – then a WUO login is performed and if successful, the found user tuple is enhanced in the USERS table with the incoming EAAHash.

```
UPDATE
     USERS
SET
     EAAHASH='f5bba3c6-6240-4ccf-8048-13dbb3405192'
WHERE
     USERID='foundID'
```

If the user has no existing account the WUO's user creation process is used to create a new WUO user. The EAAHash must then be appended to the created user.

```
INSERT INTO USERS
     (NAME,…,EAAHASH)
VALUES
     ('Muster',…,'f5bba3c6-6240-4ccf-8048-13dbb3405192')
```

AddressUpdater

The AddressUpdater is a tool which can retrieve user information from a facility where the user is registered at and display it at the Umbrella website so that the user can mutate his information and submit it to all facilities.

Information Retriever

To retrieve the information, the WUO must enable a UserInformation-Endpoint, where a user logged in to the Umbrella can get a list of his information registered at this specific facility. As this usually is a cross-domain request, JSONP must be used here.

Following a code snippet in PHP which explains the functionality:

```
// retrieve the Umbrella ID from the Headers
$shibhash = $_SERVER["EAAHash"];

// make sure it is not empty
if($shibhash <> ""){

     // query for the attributes for the specific user
     $result=$db->Execute("SELECT
     USERNAME,PASSWORD,USERID,TITLE||'     '||FIRSTNAME||'
     '||MIDDLENAME||'            '||LASTNAME            AS
     FULLNAME,EMAIL,STATUS,FIRSTNAME,MIDDLENAME,LASTNAME,
     PHONE,TITLE,SEX   FROM   USERS   WHERE   EAAHASH=:p1",
     array("p1" => $shibhash) );

     // retrieve the attributes from the database
     $userid=$result->fields[2];
     $fullname=$result->fields[3];
     $useremail=$result->fields[4];
     $firstname=$result->fields[6];
     $middlename=$result->fields[7];
     $lastname=$result->fields[8];
     $phone=$result->fields[9];
     $title=$result->fields[10];
     $sex=$result->fields[11];

     // display the attributes as JSONP
     echo          $_GET["jsonp"]          ."({\"Userid\":
     \"$userid\",\"Fullname\":  \"$fullname\",  \"Email\":
     \"$useremail\",     \"Firstname\":     \"$firstname\",
     \"Middlename\":     \"$middlename\",     \"Lastname\":
     \"$lastname\",   \"Phone\":   \"$phone\",   \"Title\":
     \"$title\", \"Gender\": \"$sex\"})";
}
```

If a client calls this endpoint he will receive a JSONP answer in following format:

```
asuidfgaiiq38zrfwhsudf({
     "Userid": "2",
     "Fullname": "Mr. Bjoern Erik Abt",
     "Email": "bjoern.abt@psi.ch",
     "Firstname": "Bjoern",
     "Middlename": "Erik",
     "Lastname": "Abt",
     "Phone": "0041563103509",
     "Title": "Mr.",
     "Gender": "M"
})
```

AttributeUpdater

In order to ensure that a user 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 exists 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.

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	Umbrella generated string
B	WUO generated string.

Figure 4 explains the workflow:

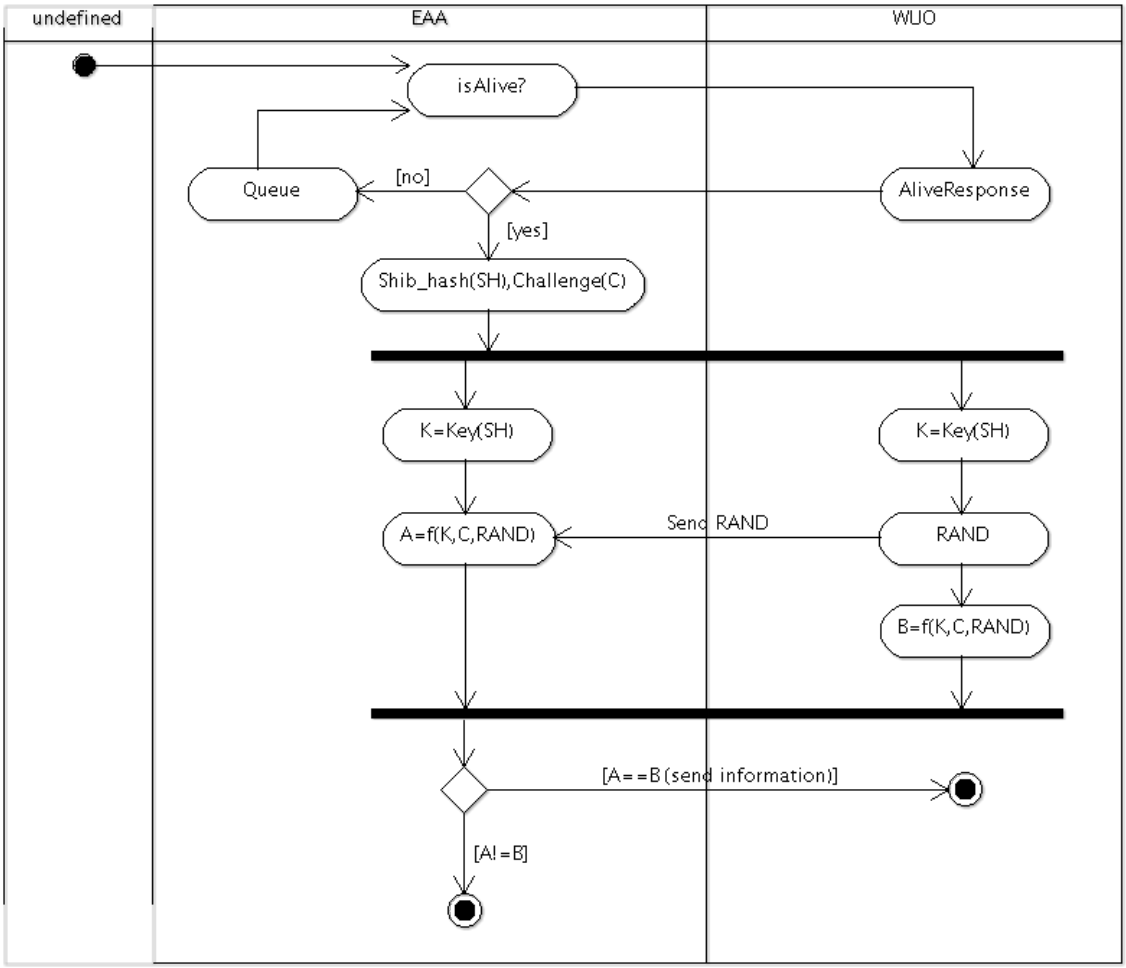


Figure 4: Challenge Response

There are existing libraries available from the Umbrella Development team.

Umbrella Account Upgrade

The Umbrella Account Upgrade functionality is a shortcut from a WUO to the Umbrella system, which assists the users in creating an Umbrella account.

The idea is to have a button inside every WUO that when clicked on it transmits the information relevant to create an Umbrella account to the Umbrella system and displays it in its registration form.

The URL to call looks as follows:

https://idp.umbrella-id.eu/euu/account/create

It accepts following POST attributes:

- username
- email
- birthdate

Graphical Elements Conditions and Status

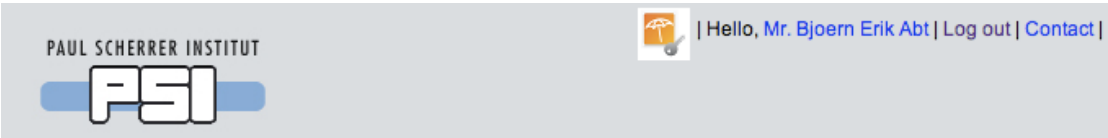
To further help users with the Umbrella system we propose following graphical helper elements depending on the condition the user is in:

Condition	Status
• User not logged in	• Login to WUO • Login to Umbrella
• Logged in at WUO • No account matching with Umbrella	• Upgrade to Umbrella
• Logged in at WUO • Accounts matched • Not logged in at Umbrella	• Login to Umbrella
• Logged in at Umbrella • Accounts matched • Logged in at WUO	• Logged-In at Umbrella

These elements should be displayed close to the user name in the WUO. As an example we show an implementation done at DUO:

Login to Umbrella

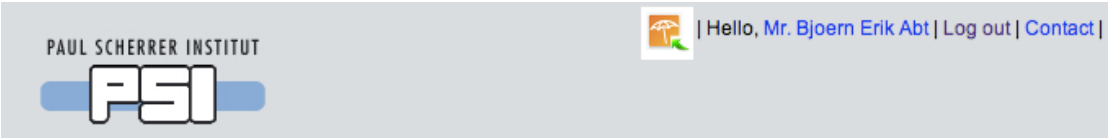
This is how it looks on a DUO system:



And the icon for itself:

Upgrade to Umbrella

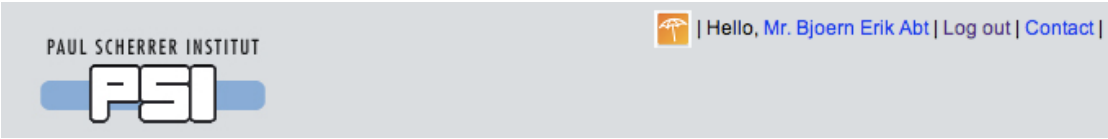
This is an example from a DUO system:



And the icon for itself:

Logged-in at the Umbrella

Again an example from the DUO system:



And the icon: