Ref. Ares(2014)3224384 - 01/10/2014

SEVENTH FRAMEWORK PROGRAMME

Scalable, Secure Storage Biobank

Grant Agreement Number: 317871

BiobankCloud Security:
D3.3, Security Toolset (alpha version)

Final

Version:	0.7
Responsible Partner:	Ali Gholami, KTH
Date:	2014-09-30

Project and Deliverable Information Sheet

Scalable Secure Storage Biobank Project	**Project Ref. №: 317871**	
	Project Title: Scalable, Secure Storage Biobank	
	Project Web Site: http://www.biobankcloud.eu	
	Deliverable ID: D3.3	
	Deliverable Nature: Report	
	Deliverable Level: **PU**	**Contractual Date of Delivery:** 30 / September/ 2014
		Actual Date of Delivery: 30 / September/ 2014
	EC Project Officer: Wolfgang Treinen / Saila Rinne	
	Partner Responsible: Ali Gholami, KTH	
	Contributing Partners: KTH & KI & Charité	

* - The dissemination levels are indicated as follows: **PU** – Public, **RE** – Restricted to other participants, **CO** – Confidential, only for members of the project (including the Commission Services).

Document Status Sheet

Version	Date	Description	Author/Partner
0.1	2014-05-14	Initial version, TOC	Ali Gholami /KTH
0.2	2104-03-06	User administration in LDAP	Ali Gholami /KTH
0.3	2014-07-29	Two-factor authentication	Ali Gholami /KTH
0.4	2014-09-10	Authorization and conclusions	Ali Gholami /KTH
0.5	2014-09-11	General comments	Lora Dimitrova/ Charité
0.6	2014-09-29	User management GUI	Jim Dowling /KTH
0.7	2014-09-30	Final version	Ali Gholami /KTH

Contents

List of Figures

EXECUTIVE SUMMARY

This deliverable presents the implementation of the security toolset alpha version in context of deliverable D3.3 for the BiobankCloud platform. The toolset contains deploying strong authentication in the platform such as two-factor mobile authentication, Yubikey tokens and public key certificates for the different categories of users. We also present the user administration tool based on OpenLDAP and GUI. The OpenLDAP supports user administration through command, while the Web application uses a GUI.

1. Introduction

This document describes the security toolset implementation according to the deliverable D3.2. In D3.2 we identified the platform security requirements and designed the security framework to be implemented by D3.3 as the alpha version.

This deliverable is structured as follows. Section 2 explains the identification scheme that is implanted in the project. Section 3 describes different authentication mechanisms that are supported. Section 4 discusses the authorization. Section 5 highlights the user administration. Section 6 presents the conclusions and future work.

2. Identification

We implemented a UNIX-based username scheme containing 8 alphanumeric characters for compliance with the KTHFS (the customized Hadoop file system). Every time a new user registers using the self-service component, a new username will be generated and stored in the backend. The username scheme is composed of 3 letters that indicate the prefix of the institution name, and the other 5 digits are the user identifiers. For example meb10003 demonstrates a user from the “MEB” institute with id number 10003. This scheme provides flexibility of user creation and integration with other institutions in cases where federation is required, in addition to high compatibility with the KTHFS.

The platform also supports the user ORCID identifiers as a replacement for the UNIX-based usernames creation. The identity store keeps mapping of ORCID identifiers and UNIX usernames for any required queries. The ORCID is supplied in addition to other necessary infomation by the users during requesting for account, as shown in Figure 1.

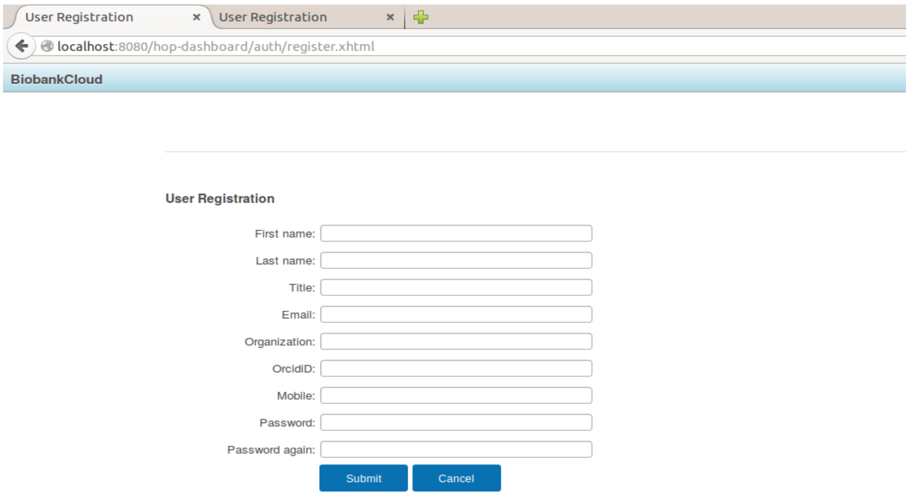


Figure 1, User registration form

3. Authentication

Requirements of using strong security mechanisms in the BiobankCloud have been discussed in the deliverables D3.1 and D3.2. We support two groups of users: users with smart mobile devices and users with Yubikey tokens, as shown in Figure 2. The motivation for this categorization stems from the fact that not all researchers use mobile devices for authentication neither they all use Yubikey tokens. Depending on the organizational policies there are different authentication mechanisms for different institutions.

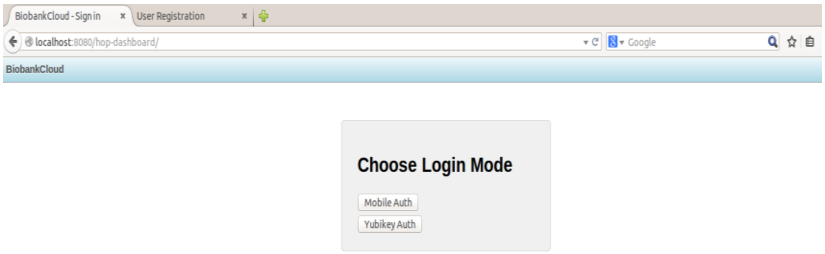


Figure 2, Selection authentication mode

In this project we developed a two-factor authentication using time-based one-time password (TOTP) for mobile devices and keyed-hash message authentication code (HMAC)-based one-time password (HOTP) for Yubikey tokens as show in Figure 3.

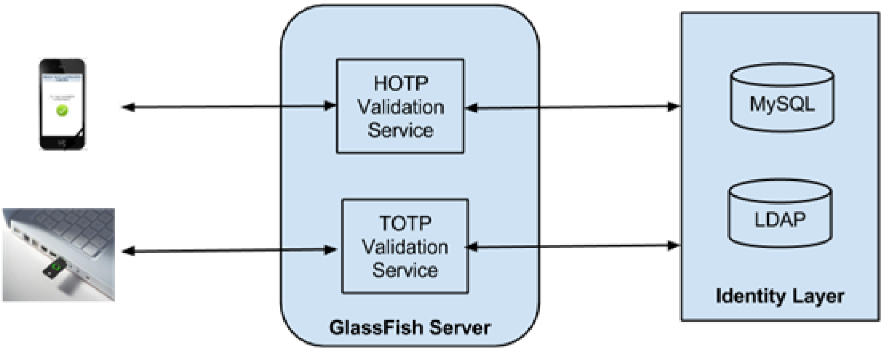


Figure 3, Two-factor authentication in the BiobankCloud

The BiobankCloud users supply the one-time password (OTP) in addition to the static passwords which are established during account creating.

3.1 Two-factor authentication using mobile devices

Google Inc. have implemented RFC6238 [4] security tokens in terms of Google Authenticator [5]. As shown in Figure 2, the Google Authenticator generates one-time passwords (six digits) that users must provide in addition to their username and password to log into Google services or other sites. The

TOTPs are generated every 30 seconds, as shown in Figure 4. We support two common mobile devices platforms: iOS and Android.

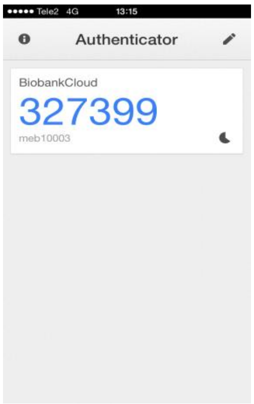
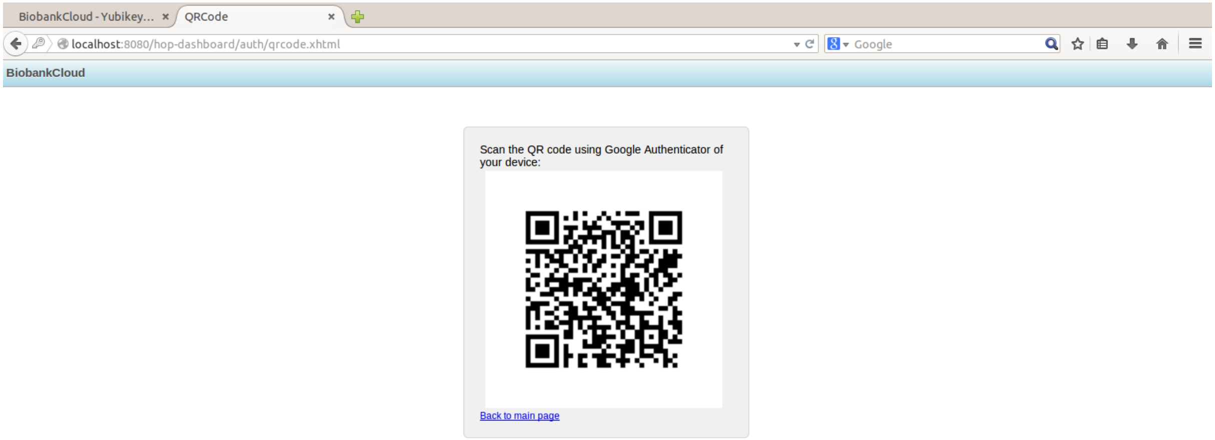


Figure 4, OTP generated by the Google Authenticator in an iPhone 5

For this aim, we implemented a customized quick response (QR) code library that contains usernames and their association information. Users only require to locate their mobile devices towards the generated QR code by the BiobankCloud and all the account information will be loaded automatically to their devices. The QR codes are presented in portable network graphics (PNG) with the size of 300 * 300 pixels, as shown in Figure 5.


BiobankCloud - Yubikey...
QRCode
localhost:8080/hop-dashboard/auth/qrcode.xhtml
Google
BiobankCloud
Back to main page

Scan the QR code using Google Authenticator of your device:

Figure 5, Generated QRs for mobile authentication

3.2 Yubikey authentication

We implemented RFC4226 [6] specification to support Yubikey authentication. We also implemented a validation service in the platform to verify the issued OTPs presented by users. A Yubikey token generates the OTPs through a push-button. Generated OTPs are sent as emulated keystrokes via the keyboard input path, thereby allowing the OTPs to be received by any text input field in the authentication page of the platform, as shown in Figure 6.

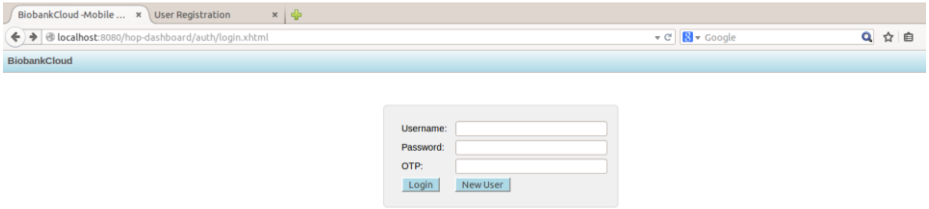


Figure 6, OTP authentication page

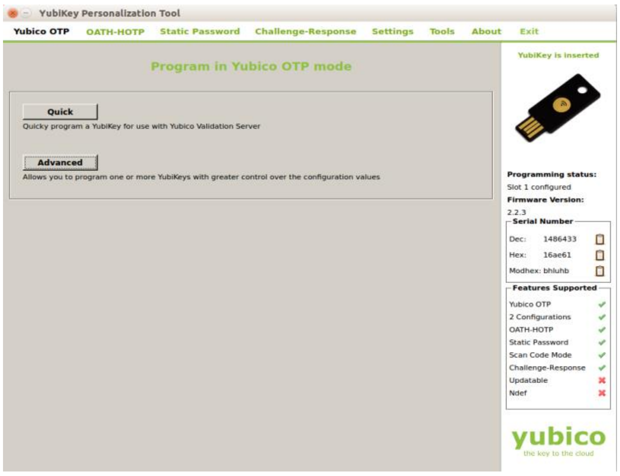
To generate an OTP, at first user inserts the Yubikey into the USB port as shown in Figure 7. The user presses the Yubikey's OTP generation button. The Yubikey generates an encrypted string of characters that are outputted as keystrokes via the keyboard port. This output will be redirected to the OTP field of the authentication page.

Figure 7, Inserting Yubikey token for configuration

The platform users are required to supply an additional static password and then press the login button to be authenticated. The Yubikey validation service is triggered and it verifies the claimed user credentials presented as a string through the Web. Our validation service converts the received string to a byte string to be decrypted using the same (symmetric) 128-bit AES key. This will be fetched from the credential stores. Then the string's checksum will be checked and if not valid, the OTP will be rejected. Additional fields will be checked as next step and if not valid, the OTP will be rejected. As next step, the non-volatile counter will be compared with the existing value in the credentials store. If lower than or equal to the stored value, the received OTP will be rejected. If greater than the stored value, the received value is stored and the OTP will be validated.

3.3 Configuring the Yubikey tokens

To configure the Yubikey, personalization GUI (Graphical User Interface) software should be installed. The command "apt-get install yubikey-personalization-gui" installs the required dependencies and it can be invoked as shown in Figure 8.



Program in Yubico OTP mode

Quicky program a YubiKey for use with Yubico Validation Server

Allows you to program one or more YubiKeys with greater control over the configuration values

YubiKey is inserted

Programming status:
Slot 1 configured
Firmware Version:
2.2.3

Serial Number

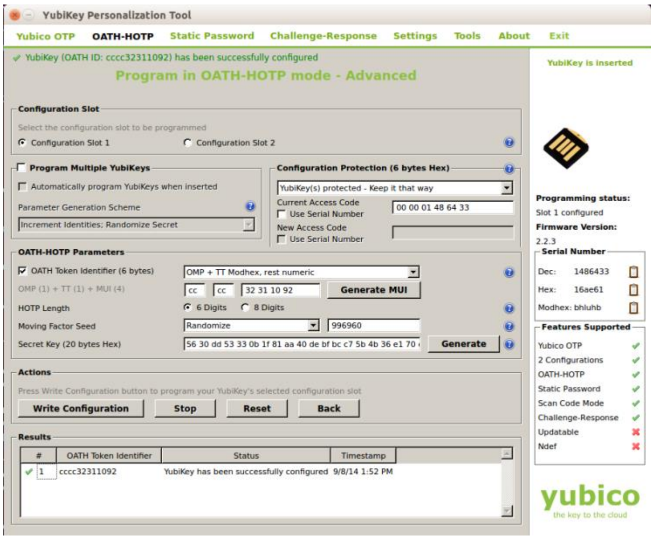
Dec:	1486433
Hex:	16ae61
Modhex:	bhluhb

Features Supported

Yubico OTP	✓
2 Configurations	✓
OATH-HOTP	✓
Static Password	✓
Scan Code Mode	✓
Challenge-Response	✓
Updatable	✗
Ndef	✗

Figure 8, Configuring Yubikey devices through Personalization tool

The credentials can be written in one of the two configuration slots as shown in Figure 9. We use OATH token identifier with 6 bytes length and also HOTP of 6 bytes length.



✓ YubiKey (OATH ID: cccc32311092) has been successfully configured

Program in OATH-HOTP mode - Advanced

Configuration Slot
Select the configuration slot to be programmed
● Configuration Slot 1 ○ Configuration Slot 2

□ Program Multiple YubiKeys
□ Automatically program YubiKeys when inserted
Parameter Generation Scheme: Increment Identities; Randomize Secret

Configuration Protection (6 bytes Hex)
YubiKey(s) protected - Keep it that way
Current Access Code: 00 00 01 48 64 33
□ Use Serial Number
New Access Code:
□ Use Serial Number

OATH-HOTP Parameters
☑ OATH Token Identifier (6 bytes): OMP + TT Modhex, rest numeric
OMP (1) + TT (1) + MUI (4): cc cc 32 31 10 92
HOTP Length: ● 6 Digits ○ 8 Digits
Moving Factor Seed: Randomize 996960
Secret Key (20 bytes Hex): 56 30 dd 53 33 0b 1f 81 aa 40 de bf bc c7 5b 4b 36 e1 70

Actions
Press Write Configuration button to program your YubiKey's selected configuration slot

Results

#	OATH Token Identifier	Status	Timestamp
✓ 1	cccc32311092	YubiKey has been successfully configured	9/8/14 1:52 PM

YubiKey is inserted

Programming status:
Slot 1 configured
Firmware Version:
2.2.3

Serial Number

Dec:	1486433
Hex:	16ae61
Modhex:	bhluhb

Features Supported

Yubico OTP	✓
2 Configurations	✓
OATH-HOTP	✓
Static Password	✓
Scan Code Mode	✓
Challenge-Response	✓
Updatable	✗
Ndef	✗

Figure 9, Configuring a Yubikey token to generate OTP

After selecting the moving factor seed as "Randomize", the "Write Configuration" action should be selected to write the configurations in both a CSV file and the actual Yubikey device. The identity store then can be updated with the following credentials:

OATH-HOTP, 9/8/14 2:02 PM, 1, ccccedebbckd, , 5630dd53330b1f81aa40debfbcc75b4b36e170ed,
000001486433,000001486433,0,1,0,6,996960,0,0,0,0,0

3.4 Authentication using public key certificates

The BiobankCloud administrative staff, such as the data controller, is required to login to the platform services using public key certificates. For this purpose, the BiobankCloud services will be identified to the clients through a server certificate and also clients require a valid certificate to communicate with each other.

The certificates are stored in the Java key store and obtaining and installing a valid certificate is done through the "keytool" command in GlassFish v3 [7]. The client distinguished name (DN) is required to be embedded in the web.xml file of the Web application.

```
 <login-config>
   <auth-method>CLIENT-CERT</auth-method>
 </login-config>

<security-role-mapping>
   <role-name>CONTORLLER</role-name>
   <principal-name>:     CN=     Ali     Gholami     nospam@kth.se,     O=Kungliga     Teknikska
Hogskolan,C=SE,DC=TCS,DC=Terena,DC=org </principal-name>
</security-role-mapping>
```

The users are then required to import the certificate to their browser when using the platform that is also protected through a password.

4. Authorization

To protect the confidentiality of the genomic data and platform services we implanted a relational table "Resource", which includes the information about a specific resource such as a directory containing the sample studies, as shown in Figure 9. All users are assigned a role when their account is activated. When users upload sample data sets, they can decide which permissions to assign to which users (read, write and execute). This is done in the "Permission" table.

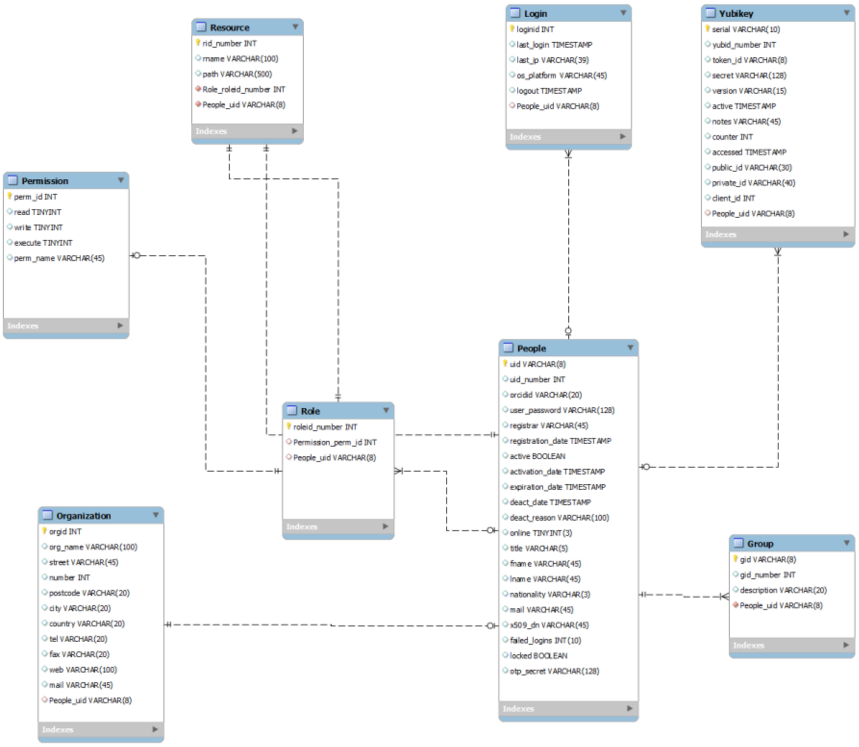


Figure 9, Authorization model to access genomic data in KTHFS

5. User administration

The BiobankCloud user administration service will be built on top of existing local administrations in different institutions and biobanks, and, hence should make no assumption about these systems. This will ensure each BiobankCloud user management system has minimum adaptation and interruption to the local systems, in addition to performing user management locally by each institution.

5.1 User administration in LDAP

To enable access to the BiobankCloud services attributes of the user have to be known by that BiobankCloud. For instance, POSIX attributes such as username, user identifier (UID), group identifier (GID) and the subject name of a X.509 certificate for some users with administrative privileges.

We implemented our directory service based on the lightweight directory access protocol (LDAP) for user administration in the BiobankCloud. LDAP provides a suitable protocol for excessive lookups compared to relational databases that are transactional. Moreover, LDAP can be used as a protocol for communication and can be integrated with several high availability databases such as the MySQL cluster.

The current setup consists of distributed LDAP servers for each BiobankCloud and each platform maintains the user information in a local LDAP server.

The LDAP servers host directory entries with suffixes on each platform such as "dc=biobankcloud,dc=eu", "dc" stands for domain component that can be different for each BiobankCloud.

The account creation in the BiobankCloud consists of two phases. First a BiobankCloud manager or administrator creates the user account in the LDAP. Second, the provision system updates the execution engine of the BiobankCloud platform (Hadoop cluster) through a provision service.

Security of the LDAP server is a major concern, and to ensure confidentiality, integrity and non-repudiation we use encryption tunnels through TLS provided by X.509 certificates.

Therefore, clients and administrative users will interact authentically using the SASL EXTERNAL mechanism. SASL provides the functionality of mapping certificate subjects to LDAP entries for authorization.

To forbid unauthorized access to the LDAP service, a list of authorized servers with their certificate subject is defined in the LDAP server. Also, we assume the LDAP service runs in a dedicated host with enough security measures that is not accessible to irrelevant programs and processes.

5.1.1 LDAP attributes, object classes and schemas

The attributes that are defined in this section, are defined based on the requirements of WP1 and WP5. However the standard attributes of LDAP were not sufficient to address the BiobankCloud requirements, and hence we defined a schema containing attributes and objectClass definitions, as described in this section.

- POSIX passwd entry: uid (login name), uidNumber, gidNumber, gecos, homeDirectory, login shell.
- POSIX group: name, gidNumber, members (memberUid).
- Title: Mr., Mrs., Ms., Miss, Dr., and Prof.
- Home organisation of the user: bbcHomeInstitute.
- Email address: mail.
- Telephone number: telephoneNumber.

- Nationality of user: bbcNationality.
- Address and telephone number of the BiobankCloud.
- The Subject Name of the admin’s certificate: The subject name (DN) of the admin users.
- Account status and deactivation reason: bbcDeactivated, bbcDeactReason.
- Name of the administrator or manager that registered the user: bbcRegistrar.
- Expiration date of the account: shadowExpire with yyymmdd or -1 values.
- Orcid Identifier: orcidID.
- Yubikey device identifier: yubikeyId.
- Yubikey static password: passwordFactor.
- Group description: Trusted Researcher, Guest Researcher, Data Provider, Auditor, Ethics Board
- Project Manager: bbcProjectManager.
- Resource: bbcProjectResource.
- Online User: bbcOnline
- Online Date: lastOnline
- OTP secret: otpSecrets

The actual implementation of the LDAP service [4], including top level entry of organization, project, group, resource and people.

```
#top level structure entry
dn: dc=biobankcloud,dc=eu
objectclass: top
objectclass: dcObject
objectclass: organization
o: biobankcloud
dc: eu

# top level orgnization entry
dn: ou=Organization,dc=biobankcloud,dc=eu
objectClass: top
objectClass: organizationalUnit

# top level user entry
dn: ou=People,dc=biobankcloud,dc=eu
ou: People
objectClass: top
objectClass: organizationalUnit

# top level group entry
dn: ou=Group,dc=biobankcloud,dc=eu
ou: Group
objectClass: top
objectClass: organizationalUnit

# top level group entry
dn: ou=Project,dc=biobankcloud,dc=eu
ou: Project
objectClass: top
objectClass: organizationalUnit

# top level resource entry
dn: ou=Resource,dc=biobankcloud,dc=eu
ou: Resource
objectClass: top
objectClass: organizationalUnit
```

```
# project entry
dn: cn=STHLM2,ou=Project,dc=biobankcloud,dc=eu
cn: STHLM2
bbcHomeInstitute: KI
bbcProjectEndTimestamp: 20140101235959Z
bbcProjectStartTimestamp: 20050101000000Z
description: STHLM2
gidNumber: 90008
memberUid: meb00001
objectClass: top
objectClass: posixGroup
objectClass: bbcProject
bbcProjectManager: meb00001

# user entry
dn: uid=meb00001,ou=People,dc=biobankcloud,dc=eu
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
objectClass: bbcUser
cn: Ali Gholami
gecos: Ali Gholami
givenName: Ali
sn: Gholami
mail: nospam@pdc.kth.se
telephoneNumber: +46 71 234500
title: Mr.
uid: meb00001
uidNumber: 1300001
gidNumber: 1300001
loginShell: /bin/bash
shadowExpire: -1
bbcNationality: SE
homeDirectory: HDFS
userpassword: 438ruifneu38ebn23r832..
orcidID: http://orcid.org/0000-1111-2222-3333
yubikeyId: 000008001261
passwordFactor: 937b0eb8ae06d881b49e5df4bf12d47c…
bbcDeactivated: FALSE
bbcDeactReason: N/A
bbcHomeInstitute: KI
shadowAccount: 20160101
bbcRegistrar: Ali Gholami
bbcSubjectDN: CN= Ali Gholami nospam@kth.se, O=Kungliga Teknikska
Hogskolan,C=SE,DC=TCS,DC=Terena,DC=org
bbcOnline:False
lastOnline:20141001,18:34
otpSecret:AES128..



# group entry
dn: cn=meb00002,ou=Group,dc=biobankcloud,dc=eu
objectClass: top
objectClass: posixGroup
cn: meb00002
gidNumber: 1300002
description: Trusted Researcher
```

```
# organization entry
dn: ou=KI,ou=Organization,dc=biobankcloud,dc=eu
objectClass: top
objectClass: organizationalUnit
ou: KI
telephoneNumber: +46 8 111 2222
postalAddress: Nobelway 1, 100 44, Solna, Stockhlom, Sweden

# resource entry
dn: rn=audit,ou=Resource,dc=biobankcloud,dc=eu
objectClass: top
objectClass: bbcProjectResource
rn: audit
```

5.1.2 The setup of the BBC LDAP server for user administration

We implemented our LDAP service using OpenLDAP on an Ubuntu 12.x platform. To install and configure OpenLDAP you should follow the following instructions or use the chef recipes through Vagrant [https://github.com/hopstart/hop-chef].

To create TLS certificates follow the instructions under TLS section of [5].

```
$sudo apt-get install slapd ldap-utils libdb5.1-dev
$sudo apt-get install gnutls-bin ssl-cert
$sudo sh -c "certtool --generate-privkey > /etc/ssl/private/cakey.pem"
$sudo certtool --generate-self-signed --load-privkey /etc/ssl/private/cakey.pem --template /etc/ssl/ca.info --outfile /etc/ssl/certs/cacert.pem
$cd /etc/ldap
$sudo certtool --generate-privkey --bits 1024 --outfile ldapkey.pem
$sudo certtool --generate-certificate --load-privkey ldapkey.pem --load-ca-certificate /etc/ssl/certs/cacert.pem --load-ca-privkey /etc/ssl/private/cakey.pem  --template /etc/ssl/ldap01.info  --outfile ldapcert.pem
$sudo service slapd restart

$cat > certinfo.ldif
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ldap/ssl/cacert.pem

add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/ssl/ldapcert.pem

add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/ssl/ldapkey.pem

$sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f certinfo.ldif
```

Check to see if LDAP service is running:
Place bbc.schema in the current directory. Convert the schema to a dynamic configuration.

```
$slaptest -f test.conf -F ldap/
```

This will generate a new schema called {4}bbc.ldif.

Replace the below string in the headers (a sample is located under [4]):

```
--
dn: cn=bbc,cn=schema,cn=config
objectClass: olcSchemaConfig
cn: bbc
--
```

Also remove the lines after “structuralObjectClass: olcSchemaConfig” at the bottom.

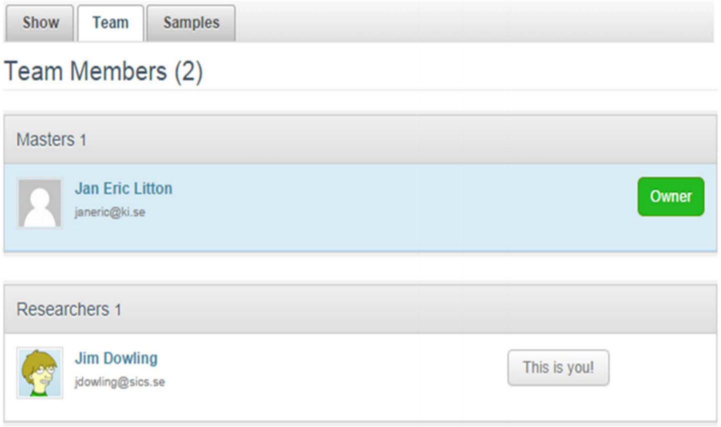
```
$sudo ldapadd -Y EXTERNAL -H ldapi:///  -f cn\=\{4\}bbc.ldif
```

To check if schema is added:

```
$sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=schema,cn=config
$sudo ldapadd -x -D cn=admin,dc=biobankcloud,dc=eu -w biobankcloud -f bbc_structure.ldif
$sudo ldapadd -x -D cn=admin,dc=biobankcloud,dc=eu -w biobankcloud -f bbc_orgs.ldif
$sudo ldapadd -x -D cn=admin,dc=biobankcloud,dc=eu -w biobankcloud -f bbc_group.ldif
$sudo ldapadd -x -D cn=admin,dc=biobankcloud,dc=eu -w biobankcloud -f bbc_resource.ldif
$sudo ldapadd -x -D cn=admin,dc=biobankcloud,dc=eu -w biobankcloud -f bbc_people.ldif
```

5.2 User administration through the Web GUI

The user administration through GUI provides the functionality to search team members belong to study samples, as shown in Figure 10.



Team Members (2)

Masters 1

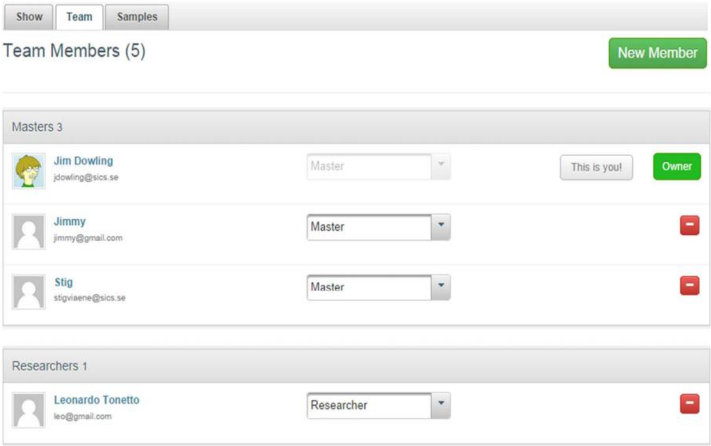
Jan Eric Litton
janeric@ki.se

Researchers 1

Jim Dowling
jdowling@sics.se

Figure 10, User administration page

A study sample owner then has enough privileges to entitle different roles to a team that require access to that specific data set, as shown in Figure 11.



Team Members (5)

Masters 3

Jim Dowling jdowling@sics.se Master

Jimmy jimmy@gmail.com Master

Stig stigviaene@sics.se Master

Researchers 1

Leonardo Tonetto leo@gmail.com Researcher

Figure 11, User entitlement page

6. Conclusions and Future Work

In this deliverable we presented the implementation of the security toolset alpha version for the BiobankCloud including two-factor authentication using mobile devices and Yubikey tokens in addition to public key certificates. We also presented user administration tools through OpenLDAP and web GUI.

However, there were some limitations for BiobankCloud federation when using GlassFish since Shibboleth does not support GlassFish. For the next deliverable D3.4, we will finalize the auditing tools in the platform and also enhance the existing security services according to feedback from WP1 and WP5.

References

[1] BiobankCloud- STREP Proposal, Call Identifier: FP7 ICT-2011-8.
[2] Deliverable D3.2, Security toolset design, WP3, 2014-06.
[3] Open Researcher and Contributor ID (ORCID) Structure, http://support.orcid.org/knowledgebase/articles/116780-structure-of-the-orcid-identifier.
[4] TOTP: http://tools.ietf.org/html/rfc6238.
[5] The Google Authenticator, https://code.google.com/p/google-authenticator/.
[6] HOTP: https://tools.ietf.org/html/rfc4226.
[7] http://docs.oracle.com/cd/E19857-01/820-1646/abxfh/index.html