Horizon 2020
European Union Funding
for Research & Innovation

C3-Cloud

"A Federated Collaborative Care Cure Cloud Architecture for Addressing the Needs of Multi-morbidity and Managing Poly-pharmacy"

PRIORITY Objective H2020-PHC-25-2015 - Advanced ICT systems and services for integrated care

D6.3 Open Source Privacy and Security Toolkits for the C3-Cloud Architecture

Work Package:	WP6 Interoperability Middleware
Due Date:	31 October 2017
Actual Submission Date:	31 October 2017
Project Dates:	Project Start Date: 01 May 2016
	Project End Date: 30 April 2020
	Project Duration: 48 months
Deliverable Leader:	SRDC

Project funded by the European Commission within the Horizon 2020 Programme (2014-2020)		
Dissemination Level		
PU	Public	Demonstrator
CO	Confidential, only for members of the consortium (including the Commission Services)	
EU-RES	Classified Information: RESTREINT UE (Commission Decision 2005/444/EC)	
EU-CON	Classified Information: CONFIDENTIEL UE (Commission Decision 2005/444/EC)	
EU-SEC	Classified Information: SECRET UE (Commission Decision 2005/444/EC)	

Document History:

Version	Date	Changes	From	Review
v0.1	03-10-2017	Template of the document associated to the demonstrator. Centralized implementation of the SPS is presented.	SRDC	WP6 Team
v0.2	05-10-2017	Improved implementation details of the SPS.	SRDC	SRDC
v0.3	06-10-2017	Added Section 5 – description of the demonstrator.	SRDC	SRDC
v0.4	07-10-2017	Added text for integration of SPS with pilot sites and C3-Cloud components.	SRDC	WP6 Team
v0.5	13-10-2017	Review by ORU	ORU	SRDC
v0.6	13-10-2017	Review by RJH	RJH	SRDC
V0.7	24-10-2017	Final version for submission	SRDC	All Consortium
V1.0	31-10-2017	Final review and editing (Sarah Lim Choi Keung & Theodoros N. Arvanitis)	WARWICK	

Contributors (Benef.)	Mustafa Yuksel, Gokce Banu Laleci Erturkmen, Bunyamin Sarigul (SRDC) Matias Wurschmidt-Wang, Anders Lönn (RJH) Nico González López (OSAKIDETZA) Phil Johns (SWFT) Gunnar Klein, Liron Karni (ORU)			
Responsible Author	Mustafa Yuksel		**Email**	mustafa@srdc.com.tr
	Beneficiary	SRDC	**Phone**	+903122101763

EXECUTIVE SUMMARY

WP6 is responsible for the Interoperability Middleware design and development. The provided solution addresses technical, semantic and privacy/security interoperability challenges to seamlessly integrate with the existing health care, social care and home/community care information systems, in order to enable patient-centric interoperable care coordination in an informed manner with the involvement of all stakeholders.

Task 6.3 is concerned with establishing the security and privacy architecture, in close relation with the pilot sites, according to the deliverables D8.1 Requirements and Use Cases of C3-Cloud Pilot Application, D3.2 Requirements Specification of the C3-Cloud Architecture, and the Description of Action. Task 6.3 started in month 9 (1 January 2017) and has ended in month 18 (31 October 2017).

This deliverable D6.3 is describing the development of a demonstrator of the Security and Privacy Toolkits for C3-Cloud. This document defines the objectives of the task, by referencing to the requirements identified in the first year of the project. The document mainly provides a description of the implementation strategies for C3-Cloud Security and Privacy Suite. It also, briefly introduces ongoing integration efforts with existing systems in the three pilot regions. This document also includes a manual for the tool. The main purpose of the demonstrator is to show the progress of the implementation of the C3-Cloud security and privacy components, in order to implement the use cases in a concrete way. The software demonstration will be given at the project review on 8 December 2017.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. DOCUMENT OVERVIEW

1.1. Purpose

The purpose of this task has been to develop a C3-Cloud Security and Privacy Suite (SPS), based on a set of open source toolkits. The development of this, together with the C3-Cloud specific interfaces and functions is described in this deliverable. The document also introduces the ongoing integration efforts with the existing security solutions in the three pilot sites. A demonstrator has been built to test the SPS and presented in the document.

1.2. Outline of the deliverable

The report on the C3-Cloud Security and Privacy Suite (SPS) demonstrator is organized as follows:

- Section 2 summarizes the basic objectives of the task and tools that must be demonstrated by referencing requirements (deliverable D3.2) [D3.2].
- Section 3 describes the implementation of C3-Cloud Security and Privacy Suite, by providing the details of its components.
- Section 4 presents the integration work, which has been carried out so far, with pilot sites and the rest of the C3-Cloud software components.
- Section 5 provides information on how to use the demonstrator in the specific example designed to address the requirements of the selected use cases.
- Section 6 addresses some future plans to continue the integration work with the C3-Cloud components and pilot sites.

1.3. The context of the Security and Privacy Suite in the C3-Cloud Component Diagram

Security and Privacy Suite (SPS) is responsible for authentication and authorisation of Care Team Members, while they are managing personalised care plans of patients and accessing sensitive personal data. SPS is also ensuring that all data exchange within and across C3-Cloud software components is encrypted and properly auditable.

In the C3-Cloud architecture (Figure 1), the patient's electronic health records received from the local EHR systems via the Technical Interoperability Suite (TIS), patient reported observations from the Patient Empowerment Platform (PEP), and the care plan of the patient managed through Coordinated Care and Cure Delivery Platform (C3DP) are all managed in the C3-Cloud FHIR Repository. Hence, each of these client apps, i.e. TIS, PEP and C3DP needs to be authenticated and authorized to access (read, write, update) patient data to C3-Cloud FHIR Repository, via the functionalities provided by the SPS. All such operations need to be logged for ensuring accountability via SPS.

It should be noted that patient authentication into the Patient Empowerment Platform (PEP) is not within the scope of SPS, but managed by the PEP itself. SPS is solely responsible for care team members' authentication and authorization among the person users, and also for authentication and authorization of all software components to access C3-Cloud FHIR Repository. Patient authentication will be presented in D5.3 - Responsive Multi-Channel Patient Empowerment Platform.

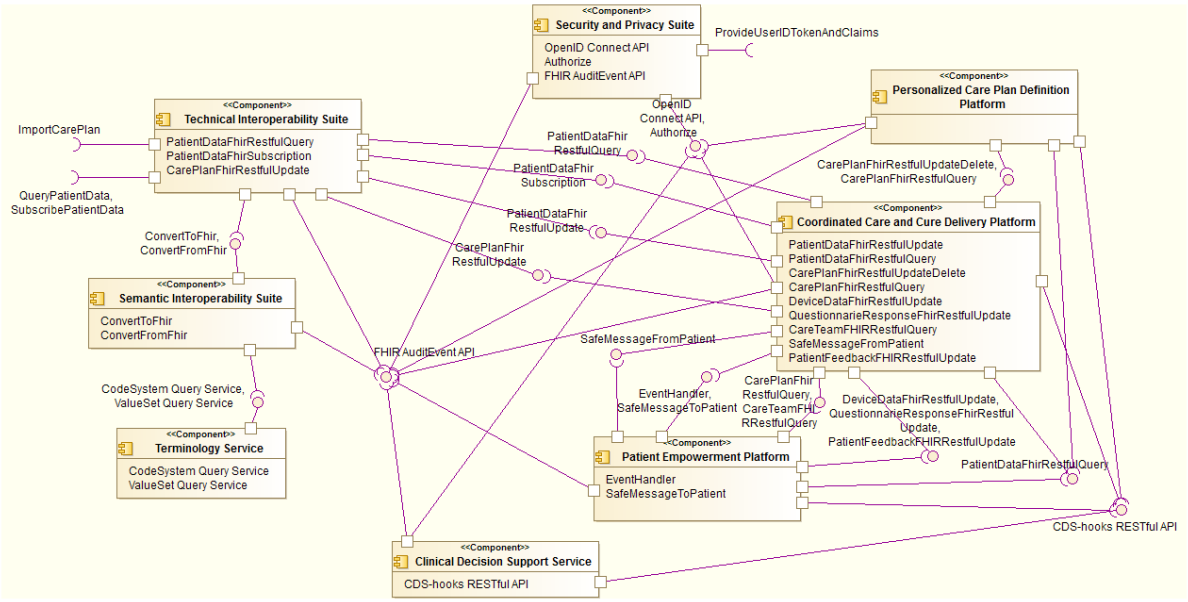

Figure 1 C3-Cloud Component Diagram from D3.3 - Conceptual Design of the C3-Cloud Architecture

1.4. Abbreviations and Acronyms

Abbreviation / Acronym	Definition
ATNA	Audit Trail and Node Authentication
C3DP	Coordinated Care and Cure Delivery Platform
CSRF	Cross-Site Request Forgery
CRUD	Create, Read, Update, Delete
EHR	Electronic Healthcare Records
GP	General Practitioner
FHIR	Fast Healthcare Interoperability Resources
HTTP	Hypertext Transfer Protocol
IdP	Identity Provider
JWT	JSON Web Token
LDAP	Lightweight Directory Access Protocol
MS AD FS	Microsoft Active Directory Federation Services
PEP	Patient Empowerment Platform
PHI	Personal Health Information
RJH	Region Jämtland Härjedalen
SAML	Security Assertion Markup Language
SIS	Semantic Interoperability Suite
SPS	Security and Privacy Suite
STU	Standard for Trial Use

SWFT	South Warwickshire NHS Foundation Trust
TIS	Technical Interoperability Suite
UI	User Interface
XSRF	Cross-Site Request Forgery

2. SECURITY AND PRIVACY SUITE OBJECTIVES

2.1. Use Cases for Security and Privacy Suite

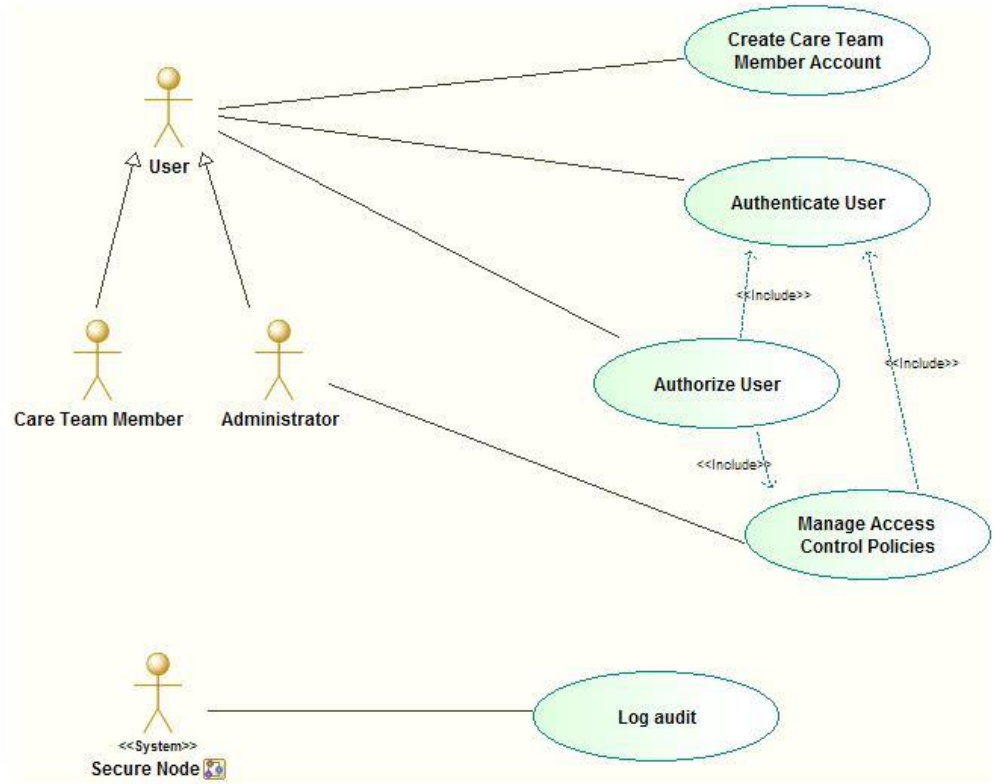

Figure 2 Use cases of the Security and Privacy Suite

Five use cases, as shown in figure 2, have been identified for the Security and Privacy Suite of C3-Cloud in D3.2 [D3.2]. These are:

- SPS-1: Create Care Team Member Account: This use case specifies the need for creating a user account for a new Care Team Member, who is not member of an organisation identity provider system, which is already integrated with the C3-Cloud applications. This should be used for login to the Coordinated Care and Cure Delivery Platform (C3DP). Normally, Care Team Members such as GPs, specialists or nurses shall continue using their business user accounts thanks to the integration to be achieved between their organisation's identity provider system (e.g., LDAP, Active Directory) and C3-Cloud Security and Privacy Suite. Those users shall not need to obtain new user accounts. Therefore, this use case is for supporting possible Care Team Members that do not have such business accounts, or when their organisation's identity provider is not integrated with C3-Cloud.
- SPS-2: Authenticate User: This use case sets the requirements to support authentication of a Care Team Member or an Administrator while signing in to the C3-Cloud system to use the Coordinated Care and Cure Delivery Platform (C3DP). Through enabling a single sign-on mechanism, the users are enabled to use C3-Cloud applications using a single account, which

is indeed the business account they use daily in their organisations (exceptional users are registered through the previous use case). Note that Patient user account management and authentications are handled through the Patient Empowerment Framework.

- SPS-3: Authorise User: This use case ensures that a Care Team Member or an Administrator is authorised to perform a specific CRUD (Create, Read, Update, Delete) operation on a specific patient resource, such as updating the care plan of a patient through the Coordinated Care and Cure Delivery Platform (C3DP).
- SPS-4: Manage Access Control Policies: This use case enables the Administrator to manage access control policies through the Authorisation Manager of the C3-Cloud Security and Privacy Suite. These policies are applied within the Coordinated Care and Cure Delivery Platform (C3DP) while granting authorisations to users for specific operations. Permission definitions shall be role-based (e.g., nurse, GP, specialist) and assigned to types of resources (e.g., care plan, goal, medication request, appointment) and operations (e.g., create, read, update, delete).
- SPS-5: Log Audit: This use case is enabling auditing of all kinds of interactions between any data provider system and data requester system in the overall C3-Cloud environment.

The details of these Use Cases are described in D3.2.

2.2. Requirements for the Security and Privacy Suite

In D3.2, high level specifications of SPS are provided in Section 4.4. These expected features of the Security and Privacy Suite (SPS) can be summarized as:

- Creating a user account for a new Care Team Member without an account linked with the C3-Cloud system
- Authenticating a Care Team Member or an Administrator and starting a secure session for him/her in C3-Cloud applications
- Guaranteeing that no unauthorised user is able to access or modify sensitive data
- Providing a dynamic mechanism for management of the access control policies, instead of static rules hard-coded in application source code
- Ensuring that all data exchange between C3-Cloud applications can be audited appropriately.

In addition to this, several functional and non-functional requirements have been defined for SPS in Section 4.4 of D3.2, which are also maintained in the Requirement Traceability Matric managed by WP3.

3. IMPLEMENTATION OF THE C3-CLOUD SECURITY AND PRIVACY SUITE

3.1. Architectural Design of the Security and Privacy Suite

In D3.3, the initial architectural design of Security and Privacy Suite is depicted as in Figure 3 [D3.3].

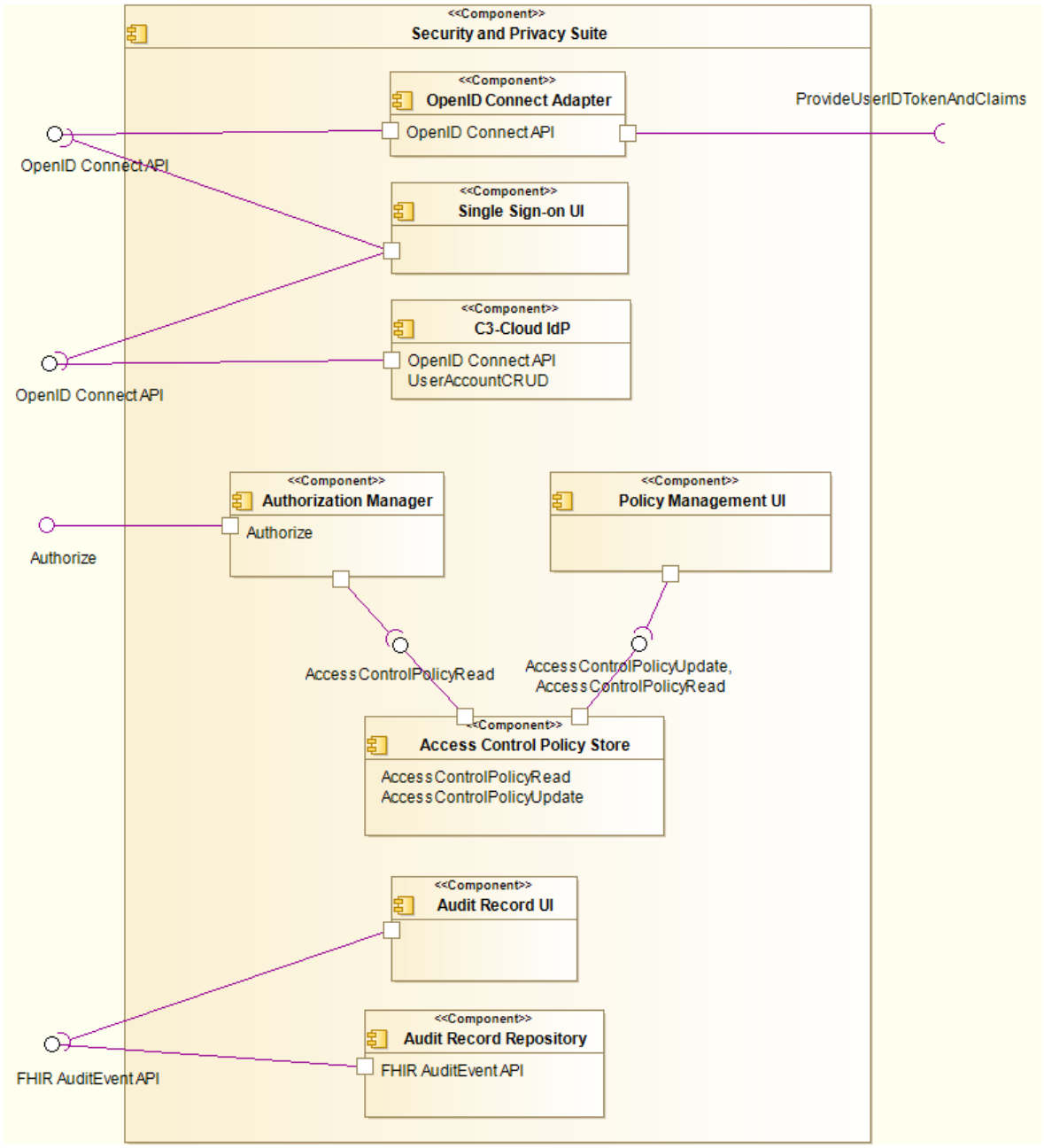

Figure 3: SPS Component Diagram in D3.3

During the implementation phase, this architectural design has been slightly modified, where server-side functionalities for authentication and authorization have been grouped under C3-Cloud SPS Server, while user interfaces related with several different features of the Security and Privacy Suite are grouped under C3-Cloud SPS Manager. On top of this, Audit Record Repository is served via the C3-Cloud FHIR Repository. This new architectural design is depicted in Figure 4.

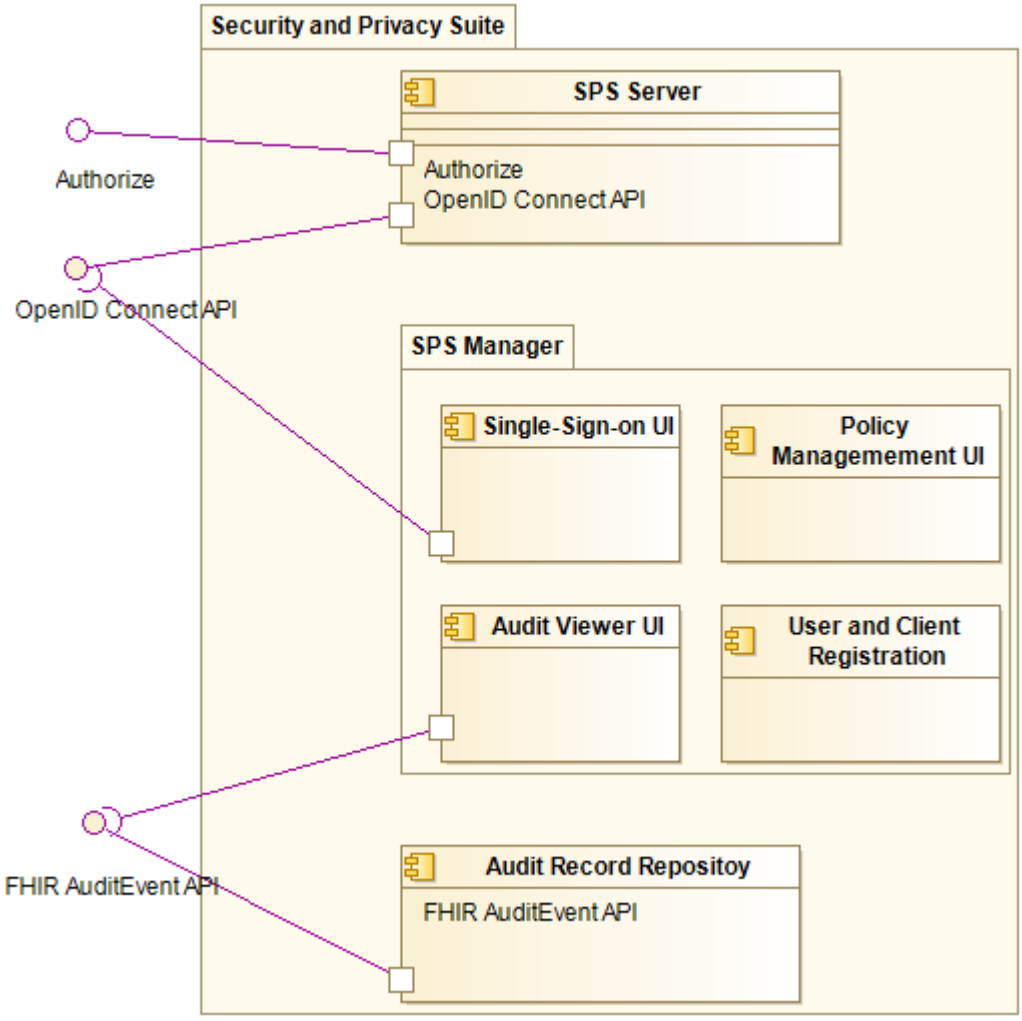

Figure 4 The new Architectural design of C3-Cloud SPS

Based on this new design C3-Cloud SPS is composed of three main components: focused on three security services:

- **C3-Cloud SPS Server** provides services for user registration, privacy policy management and endpoints defined in **OpenID Connect 1.0** standard to perform authentication and authorization (Authorization Endpoint, Token Endpoint, etc.). By implementing OpenID Connect API, it serves C3-Cloud Identity Provider (IdP), and it also manages the C3-Cloud Access Control Policy Store.
- **C3-Cloud SPS Manager** a web application for representing the functionalities of C3-Cloud SPS Server with the following user screens; single sign on UIs, policy management UI, client registration UI, user registration UI and audit viewer UI.
- **Audit Record Repository** is a FHIR repository that maintains Audit Trails implemented as FHIR AuditEvent resource. In C3-Cloud architecture, the C3-Cloud FHIR Repository is used as the Audit Record Repository.

3.2. Implementation details

3.2.1. Authentication and Authorization via C3-Cloud SPS Server

Authentication and authorization in the C3-Cloud SPS Server is managed by a single sign on policy, and is implemented in compliance with OpenID Connect 1.0 Standard [OPENIDCONNECT]. OpenID Connect is a simple authentication layer built on top of the OAuth 2.0 protocol [OAUTH]. It enables

clients to verify the claimed identity of a user, as well as to obtain basic profile information about the user (including information about when the user in question is authenticated and how) in an interoperable and REST-like manner. User ID tokens and claims (user info) are represented and exchanged in the JSON Web Token (JWT) format.

In C3-Cloud, every client application (such as C3DP, PEP, TIS) should request user authorization from the C3-Cloud SPS Server to get an access token. Applications can use that token to access users' protected information stored in the C3-Cloud FHIR Repository.

Briefly, a client application should follow these steps to get an access token from the server (see Figure 5):

(1) Client Sends Authentication Request: Client application prepares an authorization request and sends it to the authorize endpoint of C3-Cloud SPS Server;

(2) User Authentication and Authorization C3-Cloud SPS Server authenticates the end user, obtains his/her consent for the client application;

(3) Authentication Response: C3-Cloud SPS Server redirects the end user back to the client application with access code;

(4) Token Request: Client application sends access code to the token endpoint of C3-Cloud SPS Server,

(5) Token Response: C3-Cloud SPS Server responds with the access token and id token;

(6 & 7) User Info Request/Response: Client uses the access token received from C3-Cloud SPS Server to access to user profile info;

(8 & 9) FHIR Request/Response: Client uses the access token received from C3-Cloud SPS Server to access to the protected resources of the user stored in C3-Cloud FHIR Repository.

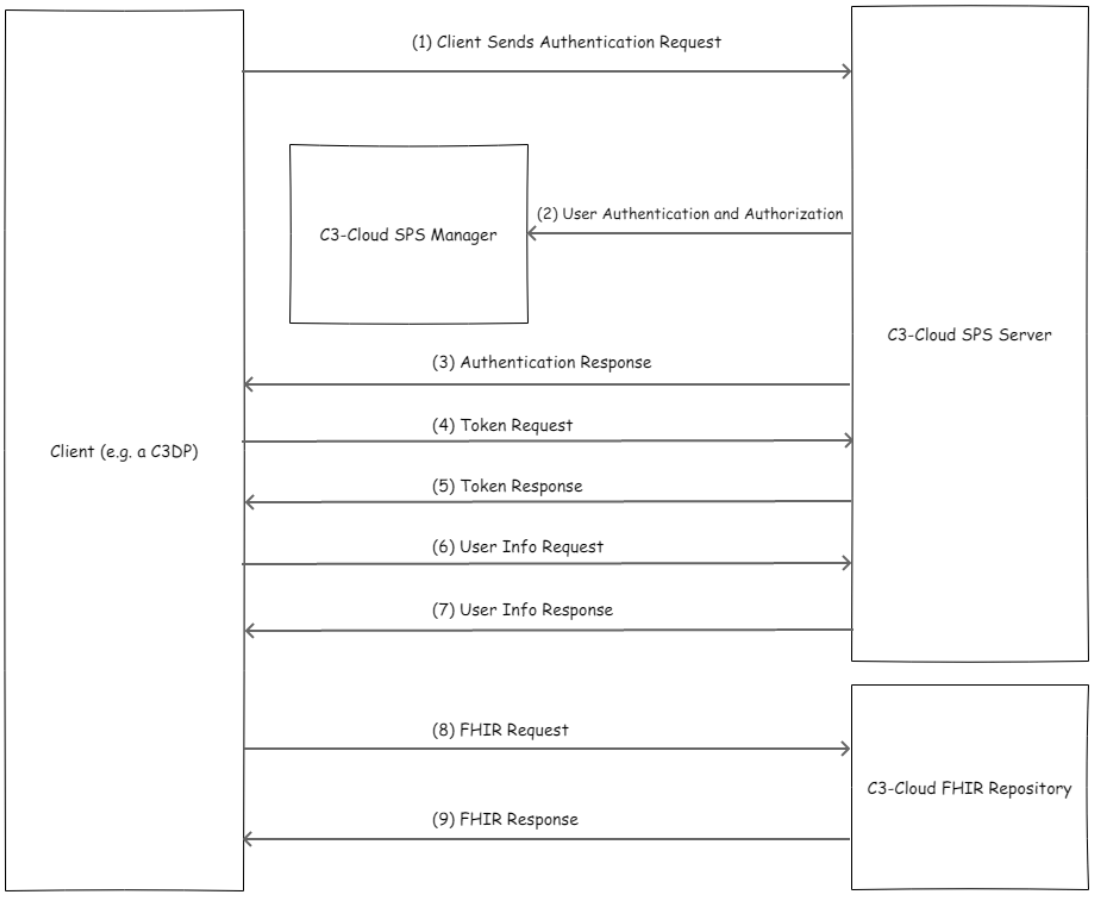

Figure 5 C3-Cloud authentication and authorization flow

The remaining part of this section will focus on the details of steps defined above. In addition to these steps, C3-Cloud SPS provides additional features for enabling authentication such as 'refreshing access tokens' and 'enabling authorization with client credentials instead of user credentials'. These are also covered in the upcoming subsections.

3.2.1.1. Client Sends Authentication Request

Client application initiates authentication flow by making a HTTP Get request to the authorize endpoint of **C3-Cloud SPS Server** with following parameters:

Table 1 Parameters of HTTP GET request (Client Sends Authentication Request)

response_type	This value must be access *code*. This type of call requests an Access Token and an ID Token be returned from the Token Endpoint in exchange for the assess code value returned from the Authorization Endpoint.
client_id	OAuth 2.0 Client Identifier valid at the Authorization Server.
scope	OpenID Connect requests MUST contain the *openid* scope value.
redirect_uri	Redirection Uri which the response will be sent. This Uri must match one of the Redirection Uri values of client.
state	Opaque value used to maintain state between the request and the call back. Typically, Cross-Site Request Forgery (CSRF, XSRF) mitigation is done by cryptographically binding the value of this parameter with a browser cookie.

If parameters provided by the client is correct, user is responded with 302 Found and redirected to the **C3-Cloud SPS Manager** for authentication. Erroneous requests are processed differently based on the error generated; If the client application provided an incorrect *client_id* or *redirect_uri*, user is redirected **C3-Cloud SPS Manager** with the proper error message, If the *client_id* and *redirect_uri* is correct then the user redirected to the client application's redirect URI address with proper error message. Example successful authentication request and response messages are depicted below:

```
GET http://app.srdc.com.tr/c3cloud/onauth/api/authorize?
  response_type=code
  &client_id=sample_client_id
  &redirect_uri=http%3A%2F%2Fapp.srdc.com.tr%2Fc3cloud%2F
               onauth%2Fsample-smart-client%2Fauth_callback.html
  &scope=openid%20profile%20patient/*.*%20offline_access
  &state=af0ifjsldkj
```

```
HTTP/1.1 302 Found
Location:
  http://app.srdc.com.tr/c3cloud/onauth/onauthmanager/login?
     scope=profile%2Bopenid%2Bpatient%2F*.*%2Foffline_access
     &redirect_uri=http%3A%2F%2Fapp.srdc.com.tr%2Fc3cloud%2F
                   onauth%2Fsample-smart-client%2Fauth_callback.html
```

```
     &client_id=sample_client_id
     &response_type=code
```

An error on *client_id* or *redirect_uri* parameters would result in:

```
HTTP/1.1 302 Found
Location:
  http://app.srdc.com.tr/c3cloud/onauth/onauthmanager/information?
    error=unauthorized_client
    &errorDesc=Invalid+client_id+parameter
```

3.2.1.2. User Authentication and Authorization

Upon successful authentication request by the client application, **C3-Cloud SPS Server** redirects user to the **C3-Cloud SPS Manager** and authentication process continues as follows:

- If the user has not been authenticated before (i.e. s/he does not already have an authenticated session), the user is redirected to the login screen and asked for his/her credentials (see Figure 6Figure 6).
- User is asked whether he/she accepts or denies client application's request for private information (see Figure 7).

3.2.1.3. Authentication Response

If user has given his/her consent to the client application, **C3-Cloud SPS Server** issues an *access code* and delivers it with the *state* parameter to the client application. Client applications are responsible for checking whether the *state* parameter matches with the *state* parameter they've sent in authentication request. On the other hand, if user denies the client application's request, user is still redirected to the client application but this time with proper error message.

For both cases, parameters to be delivered to the client application are added as query parameters to the URI of the client application using the *application/x-www-form-urlencoded* format. Example authentication response messages (one sent after an approved request and one sent after a denied request) are depicted below:

```
HTTP/1.1 302 Found
Location:
  http://app.srdc.com.tr%/c3cloud/onauth/sample-smart-client/auth_callba
ck.html?
    code=SplxlOBeZQQYbYS6WxSbI
    &state=af0ifjsldkj
```

```
HTTP/1.1 302 Found
Location:
 http://app.srdc.com.tr/c3cloud/onauth/onauthmanager/information?
   error=access_denied
   &errorDesc=User+denied+application+access
```

```
    &state=af0ifjsldkj
```

3.2.1.4. Token Request

After getting the authorization code from the query parameters, client application makes a token request by presenting its authorization grant (in the form of an *authorization_code*) to the Token Endpoint of **C3-Cloud SPS Server**. In addition, clients should provide the same *redirect_uri* parameter that they used while performing the authentication request.

The token Endpoint of **C3-Cloud SPS Server** requires client authentication depending on the type of client which is making the token request. For public clients (e.g., User-agent based web applications, native mobile applications), client authentication using *client_id* and *client_secret* is not required but they must provide their *client_id* as a parameter. Confidential clients (who can store a secret, i.e. private credential, safely) are required to authenticate themselves using HTTP Basic which is an authentication scheme which is defined in the Section 2.3.1 of the OAuth 2.0 Authorization Framework.

Requests to the Token Endpoint is made with HTTP Post request with the required parameters are added to the body of the request using *application/x-www-form-urlencoded* format. Below, two example token request messages are depicted; one by a confidential client application, and one by a public client application:

```
POST c3cloud/onauth/api/token HTTP/1.1
Host: app.srdc.com.tr
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code
&code=SplxlOBeZQQYbYS6WxSbIA
&redirect_uri=http%3A%2F%2Fyet.another.secure.server%2Fcb
```

```
POST c3cloud/onauth/api/token HTTP/1.1 HTTP/1.1
Host: app.srdc.com.tr
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code
&code=SplxlOBeZQQYbYS6WxSbIA
&client_id=fca8d49c-3f53-4886-8f7c-0cdf8a9d9a09
&redirect_uri=http%3A%2F%2Fapp.srdc.com.tr%2Fc3cloud%2F
                onauth%2Fsample-smart-client%2Fauth_callback.html
```

3.2.1.5. Token Response

Upon successful token request, the **C3-Cloud SPS Server** responds with the access token object in JSON format. Token response contains following fields:

Table 2 Details of Access Token object

access_token	Access token for accessing protected resources. This token can be used for UserInfo Endpoint and C3Cloud FHIR Repository.
scope	Scopes that this access token is authorized for. Please see Table 3 for the "scopes" defined by C3-Cloud SPS Server
id_token	See OpenID ID token (JWT token that contains Claims about the Authentication event for the End user).
refresh_token	Refresh token. See *offline_access* scope.
token_type	OAuth 2.0 Token Type value. The value is always *Bearer*.
refresh_token	OPTIONAL. This scope value is used to access the UserInfo of patient (only given_name, family_name, picture, gender, birthdate). Users also may get this scope with patient/*.*.
expires_in	Expiration time of the access token in seconds since the response was generated.

If a token request results with an error, a proper error code and error description is returned to the client application in JSON format. The following is an example for successful token response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache, no-store
Pragma: no-cache

{
  "access_token": "SlAV32hkKG",
  "token_type": "Bearer",
  "scope": "patient/Observation.write patient/UserInfo.read…",
  "refresh_token": "8cjhj43d5893jic4dg4c22bh10b8i1",
  "expires_in": 3600,
  "id_token": "eyJraWQ....bePk2ob2tJFJ....iZz5Og",
  "patient": "00d545e8-7c86-4c16-84bc-7397f6e8741e"
}
```

The following is an example for erroneous token response;

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json
Cache-Control: no-cache, no-store
Pragma: no-cache

{
```

```
  "error": "invalid_grant"
  "errorDesc": "Only authorization_code is supported."
}
```

Table 3 Scope values defined in C3-Cloud SPS Server

openid	Informs the Authorization Server that the Client is making an OpenID Connect Authentication request.
profile	This scope value requests access to the End-User's default profile information (claims), which are: given_name, family_name, preferred_username, picture, gender, birthdate, zoneinfo, locale, c3c_role, c3c_organization (See Table 4for the definition of default profile information for users).
patient/resourceType.permission	This scope value requests access to the patient's protected resources on C3-Cloud FHIR Repository. The resource type could be any FHIR resource type or (*) (which means all resource types the user authorized for) and the permission could be read, write or (*) (which means both read and write access). E.g., patient/Observation.read means read all observations about patient, patient/*.read means read all available data about patient.
user/resourceType.permission	This scope value requests access to the all patient's protected resources on C3-Cloud FHIR Repository that this user is authorized to access. The resource type could be any FHIR resource type or (*) (which means all resource types the user authorized for) and the permission could be read, write or (*) (which means both read and write access). In C3-Cloud, all of the pilot sites specified a policy that enables the authenticated healthcare professionals involved in C3-Cloud pilots, to access the data of all of the patients involved in the C3-Cloud pilot in that pilot site. Hence, this scope will be used to enable this.
patient/UserInfo.permission	OPTIONAL. This scope value is used to access the UserInfo of patient (only given_name, family_name, picture, gender, birthdate). Users also may get this scope with patient/*.*.
fhir/patient	This scope is defined for confidential clients that are authorized with client credentials and that needs offline access to C3-Cloud FHIR Repository (without any user authentication). They can request this scope to access all the data in C3-Cloud FHIR Repository. In C3-Cloud, TIS will request and use this scope to read/write the records of all of the patients involved in C3-Cloud pilot into C3-Cloud FHIR Repository.
offline_access	This scope value requests that an OAuth 2.0 Refresh Token be issued that can be used to obtain an Access Token that grants access to the End-User's UserInfo Endpoint even when the End-User is not present (not logged in).

Table 4 Default profile information for users in C3-Cloud SPS Server

sub	**Subject (user) identifier.**
name	Full name of user.
preferred_username	Username of user.
email	Email address of the user, which is linked with his identity at the issuer.
zoneinfo	Time zone of the user.
locale	User's locale, represented as language tag (e.g., en-US, fr-CA).
phone_number	Preferred phone number of the user.
c3c_role	Structural role of the user (e.g., physician, nurse, social care worker). This is a C3-Cloud extension.
c3c_organization	The organization that the user is working for. This is a C3-Cloud extension.

3.2.1.6. Client Uses Token to Access Information

Client applications can use access token received from **C3-Cloud SPS Server** to access UserInfo Endpoint and C3-Cloud FHIR Repository resources by inserting the access token to the authorization header of the HTTP requests. Following sections exemplify such a request made to the secure endpoints.

a. User Info Request and Response

An example UserInfo request and response:

```
GET c3cloud/onauth/api/token HTTP/1.1
Host: app.srdc.com.tr
Authorization: Bearer SlAV32hkKG
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache, no-store
Pragma: no-cache

{
  "sub": "294d92b6-7610-40a3-a3f2-44443486d1f2"
  "name": "Martha",
  "surname": "Cook",
  "preferred_username": "martha_cook",
  "c3c_role": "GP",
  "c3c_organization": "osakidetza"
}
```

b. Client Accesses C3-Cloud FHIR Repository Resources (FHIR Request/Response)

C3-Cloud FHIR Repository checks the reliability of access tokens by introspecting them from **C3-Cloud SPS Server.** Rather than reliability, token introspection also gives information to the C3-Cloud FHIR Repository about the token. For example, even though a client application tries to run a query on all Observation records, C3-Cloud FHIR Repository limits the query only to the patients that the access token is authorized for. As a result the query will return the data that the user has access rights to receive. Following is such a search request and response for Observation records:

```
GET /c3cloud/fhir-secure/Observation

Host: app.srdc.com.tr

Authorization: Bearer SlAV32hkKG
```

```
HTTP/1.1 200 OK

Content-Type: application/json

{

 "resourceType": "Bundle",

 "id": "ffc3a75a-32a5-435a-9597-e2c8c7b64c95",

 "type": "searchset",

 "total": 1,

 "link": [

  {

    "relation": "self",

    "url": "http://app.srdc.com.tr/c3cloud/fhir-secure/Observation?patie
nt=00d545e8-7c86-4c16-84bc-7397f6e8741e"

  }

 ],

 "entry": [

  {

   "fullUrl": "http://app.srdc.com.tr/c3cloud/fhir-secure/Observation/14
a94454-410a-4e93-8ee5-691460c7bba5",

   "resource": {

    "resourceType": "Observation"…

   }

  }

 ]

}
```

3.2.1.7. Refreshing Access Tokens

Client applications can use the *expires_in* field from the authorization response to determine when its access token will expire. After an access token expires, it may be possible to request an updated token

without user intervention, if the app asked for a refresh token via the *offline_access* scope and the server supplied a *refresh_token* in the authorization response. To obtain a new access token, the app issues an HTTP POST to the Token Endpoint, with content-type *application/x-www-form-urlencoded*

For confidential clients, an Authorization header using HTTP Basic authentication is required, where the username is the app's *client_id* and the password is the app's *client_secret*. For public clients, authentication is not possible (and thus not required) but *client_id* must be provided.

A client makes a Token Request by presenting its Authorization Grant (in the form of an Refresh Token) to the Token Endpoint using the *refresh_token* value for *grant_type* parameter with refresh_token value. In addition, *scope* parameter could also be defined and if present, scope value must be a strict sub-set of the scopes granted in the original request (no new permissions can be obtained at refresh time). A missing value indicates a request for the same scopes which are granted in the original request.

The following is a non-normative example of a Token Request made by confidential client (with line wraps for the display purposes only):

```
POST c3cloud/onauth/api/token HTTP/1.1
Host: app.srdc.com.tr
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token
&refresh_token=8cjhj43d5893jic4dg4c22bh10b8i1
```

The following is a non-normative example of a Token Request made by public client (with line wraps for the display purposes only):

```
POST c3cloud/onauth/api/token HTTP/1.1
Host: app.srdc.com.tr
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token
&refresh_token=8cjhj43d5893jic4dg4c22bh10b8i1
&client_id=fca8d49c-3f53-4886-8f7c-0cdf8a9d9a09
```

A successful *refresh_token* response is same with the *access_token* response except *id_token* may not be present at the body. If a new refresh token is present at the response, clients should replace the old refresh token with the new one.

3.2.1.8. Authorization with Client Credentials

C3-Cloud also has some applications that require access to the private information but have not any UI elements (e.g., TIS). Such client applications may request access tokens by authenticating themselves with HTTP Basic scheme. To obtain an access token using client credentials, client makes a request to the Token Endpoint using client_*credentials* value for the *grant_type* parameter. The following is an example of such a token request and response:

```
POST c3cloud /onauth/api/token
```

```
Host: app.srdc.com.tr
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

grant_type=client_credentials
&scope=fhir/patient
```

```
HTTP/1.1 200 OK
Cache-Control: no-cache, no-store
Pragma: no-cache
{
  "access_token": "jfieej3hic5gj55ai50cded6ag3b5j",
  "token_type": "Bearer",
  "scope": [
    "fhir/patient"
  ],
  "expires_in": 3600
}
```

3.2.2. User and Client Registration via C3-Cloud SPS Server

3.2.2.1. User Registration

C3-Cloud SPS Server also provides endpoints for creating and updating users. The endpoint only accepts requests if a valid access token that is authorized for required scopes is provided at the Authorization header. Following sections explains the details of those endpoints.

a. Create Endpoint

In order to perform user registration, the authenticated end user (that will perform the user registration) should be authorized for the at least one *user_registration/{role}* scope. In C3-Cloud, users who have *group_admin* role can register users either with *practitioner*, *nurse* or *social_care_worker* role. The schema and details of registration request is defined below:

```
POST c3cloud/onauth/api/userinfo
Host: app.srdc.com.tr
Authorization: Bearer access_token
{
  username: String,
  password: String,
  family_name: String,
  given_name: String,
```

```
middle_name: String,
picture: String,
gender: String,
birthdate: String,
c3c_role: String,
zone_info: String,
email: String,
address: Address Claim object,
locale: String,
}
```

Table 5 Details of User Registration Request

username	**REQUIRED. Preferred username of the user which will be used for login**
password	REQUIRED. User password.
given_name	REQUIRED. Name of the user.
family_name	REQUIRED. Surname/Family name of the user.
middle_name	OPTIONAL. Middle name(s) of the user.
gender	REQUIRED. Gender of the user.
birthdate	REQUIRED. Birthdate of the use (in ISO date format e.g. 1966-03-03).
email	OPTIONAL. Email of the user.
address	OPTIONAL. JSON Object representing the address of the user (See Section 5.1.1 of **OpenID Connect Core 1.0** for the details of address claim).
picture	OPTIONAL. URL of user's profile picture.
zone_info	OPTIONAL. User's time zone.
locale	OPTIONAL. Language tag (e.g. en-US) indicating the language of user.
c3c_role	REQUIRED. The indicator for the type of the user (practitioner | nurse | social_care_worker).

For performing registration, the parameters defined above should be wrapped in a JSON object and sent to the UserInfo endpoint with HTTP POST request with access token that is authorized for registration at the Authorization header. Following is an example request for registering a practitioner:

```
POST c3cloud/onauth/api/userinfo
Host: app.srdc.com.tr
Authorization: Bearer SlAV32hkKG
Content-Type: application/json
```

```
{
  "username": "practitioner_swft1",
  "password": "password",
  "family_name": "Jane",
  "given_name": "Doe",
  "gender": "female",
  "birthdate": "1966-10-10",
  "c3c_role": "practitioner",
  }
```

Upon successful registration request, the **C3-Cloud SPS Server** responds with 201 CREATED with stored UserInfo of the registered user, returned UserInfo may contain additional fields. The following is a non-normative example of a registration response:

```
HTTP/1.1 201 CREATED
Content-Type: application/json
Cache-Control: no-cache, no-store
Pragma: no-cache

{
  "sub":"36d8b11b-1c68-421d-bc89-cc7a92471e61",
  "given_name": "Doe",
  "family_name": "Jane",
  "preferred_username":" practitioner_swft1",
  "gender": "female",
  "c3c_role": "practitioner",
  "c3c_organization":"swft"
}
```

An unsuccessful create response would be 401 Unauthorized or 400 Bad Request with either FHIR Operation Outcome (which means there was a problem at FHIR Repository) or a message that explain the error. The following is a non-normative example of a registration response (selected username is already taken):

```
HTTP/1.1 400 Bad Request
Content-Type: text/plain
Cache-Control: no-cache, no-store
Pragma: no-cache
{
  “error”: “username_taken”
```

```
  “error_desc”: “Username is already taken. Please select another username”
}
```

b. Update Endpoint

Update operations could only be performed by the owner of the UserInfo, in other words, only the user himself is authorized to update his user profile. In order to perform update operation, the user must be authorized for the user_info_update scope. Parameters of the update operation are the same for create operation except the username, password, role and c3c_role could not be updated from this endpoint. Those parameters will be ignored even if they exist in request body. The schema of update request is defined below:

```
PUT c3cloud/onauth/api/userinfo
Host: app.srdc.com.tr
Authorization: Bearer access_token
{
  family_name: String,
  given_name: String,
  middle_name: String,
  picture: String,
  gender: String,
  birthdate: String,
  zone_info: String,
  email: String,
  address: Address Claim object,
  locale: String,
 }
```

For updating profile information, the parameters defined above should be wrapped in a JSON object and sent to the UserInfo endpoint with HTTP PUT request. Following is an example request for updating profile information:

```
POST c3cloud/onauth/api/userinfo
Host: app.srdc.com.tr
Authorization: Bearer SlAV32hkKG
Content-Type: application/json
{
  "family_name": "Jane",
  "given_name": "Doe",
  "gender": "female",
  "birthdate": "1966-10-10",
```

```
}
```

Upon successful update request, **C3-Cloud SPS Server** responds with 200 OK with updated UserInfo at the body. The following is a non-normative example of an update response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache, no-store
Pragma: no-cache

{
  "sub":"36d8b11b-1c68-421d-bc89-cc7a92471e61",
  "given_name": "Doe",
  "family_name": "Jane",
  "preferred_username":" practitioner_swft1",
  "gender": "female",
  "c3c_role": "practitioner",
  "c3c_organization":"swft"
}
```

An unsuccessful update response would be 401 Unauthorized or 400 Bad Request with a message that explains the error. The following is a non-normative example of a error response (request without name parameter):

```
HTTP/1.1 400 Bad Request
Content-Type: text/plain
Cache-Control: no-cache, no-store
Pragma: no-cache
The request content was malformed:
No usable value for name
```

3.2.2.2. Client Registration

In order to use services that are served the **C3-Cloud SPS Server**, client applications must register themselves to the **C3-Cloud SPS Server**. **C3-Cloud SPS Server** uses “Dynamic Client Registration” process explained in **OpenID Connect Dynamic Client Registration 1.0** [OPENID-CLIENTREG]. Following fields are important for C3-Cloud to register a new client (for the full list of fields and their details see section 2 of **OpenID Connect Dynamic Client Registration 1.0)**:

Table 6 Details of Client Registration Request

redirect_uris	**REQUIRED. Array of Redirection URI values used by the client. One of these registered Redirection URI values MUST exactly match the *redirect_uri* parameter value used in each Authorization Request.**

grant_types	OPTIONAL. List of the grant types client is declaring that it will restrict itself to using. The grant type values used by **C3-Cloud SPS Server** are: • authorization_code: It is used for authorization process using code. • client_credentials: It is used for authorization with client credentials. • refresh_token: It is used for refreshing access token
client_name	RECOMMENDED. Name of the Client to be presented to the end-user. If present, client name is displayed to end-user during approval process. e.g., C3-Cloud Technical Interoperability Suite
logo_uri	RECOMMENDED. URL that references a logo for the Client application. If present, the logo is displayed to the End-User during approval. The value of this field must point to a valid image file.
client_uri	OPTIONAL. URL to the home page of the client application. The value of this field must point to a valid Web page. If present, this URL is displayed to the end-user during approval.
contacts	OPTIONAL. Array of e-mail addresses of people responsible for this client application. If present, contacts are displayed to the end-user during approval.
policy_uri	REQUIRED. String that points to a human-readable privacy policy document that describes how the deployment organization collects, uses, retains, and discloses personal data. The value of this field must point to a valid web page. If present, this URL is displayed to the End-User during approval.
tos_uri	OPTIONAL. URL that the client application provides to the end-user to read about its terms of service. The value of this field must point to a valid web page. If present, this URL is displayed to the end-user during approval.
token_endpoint_auth_method	OPTIONAL. Requested Client Authentication method for the Token Endpoint. The defined options are *client_secret_basic* and *none*. If omitted, the default is *client_secret_basic*. Public apps should select this option as *none*.

For performing registration, the parameters defined above should be wrapped in a JSON object and sent to the Client endpoint with Http POST request. Following is an example request for registering a client.

```
POST c3cloud/onauth/api/register
Host: app.srdc.com.tr
Content-Type: application/json
{
  "redirect_uris": [
    "http://app.srdc.com.tr/c3cloud/onauth/sample-smart-client/authorize-cb"
  ],
  "grant_types": [
```

```
    “authorization_code"
  ],
  "client_name": "SRDC Sample Client",
  "logo_uri": "http://www.srdc.com.tr/wp-content/uploads/2014/12/srdc-wp.png",
  "client_uri": "http://dummyclient.com",
  "contacts": [
    "mustafa@srdc.com.tr"
  ],
  "token_endpoint_auth_method": "none"
}
```

Upon successful registration request, **C3-Cloud SPS Server** responds with 201 CREATED with stored client metadata of the registered client application. Returned client metadata includes unique *client_id* and *client_secret* of the client (i.e. the private credentials that will be used by the client for authentication requests, see Section 3.2.1.8). Public applications which select *token_endpoint_auth_method* as *none* may ignore client secret but private applications should keep their secret safe. The following is a non-normative example of a registration response:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "client_id": "fdb74fa7-cb60-423f-87ee-89dba4c490c1",
  "client_secret": "r247v76ngrnga8d7hdojmmaj9m",
  "redirect_uris": [
    "http://app.srdc.com.tr/c3cloud/onauth/sample-smart-client/authorize-cb"
  ],
  "client_name": "SRDC Dummy Client",
  "client_uri": "http://dummyclient.com",
  "logo_uri": "http://www.srdc.com.tr/wp-content/uploads/2014/12/srdc-wp.png",
  "contacts": [
    "mustafa@srdc.com.tr"
  ],
  "token_endpoint_auth_method": "none",
  "scope": "profile openid address email patient user fhir/patient",
  "grant_types": [
    "authorization_code"
  ],
  "response_types": [
```

```
    "code"
  ],
  "application_type": "web",
  "request_object_encryption_enc": "A128CBC-HS256",
  "user_info_encrypted_response_enc": "A128CBC-HS256",
  "id_token_signed_response_alg": "RS256",
  "id_token_encrypted_response_enc": "A128CBC-HS256",
  "client_secret_expires_at": 0,
  "require_auth_time": false
}
```

An unsuccessful registration response would be 400 Bad Request with a message that explains the error. The following is a non-normative example of an error response (request without redirect_uri field):

```
HTTP/1.1 400 Bad Request
Content-Type: application/json


{
  "errorCode": "invalid_redirect_uri",
  "errorDesc": "At least 1 redirect_uri should be provided.",
}
```

3.2.3. C3-Cloud SPS Manager

The C3-Cloud SPS Manager is implemented as a web application that serves the user interfaces for the following functionalities:

- Single Sign-on & Consent/Approval
- Authorization Policy Management
- Client Registration
- User Registration
- Audit viewer

The user interfaces of the SPS Manager are implemented with Angular 2+ [ANGULAR] and using both bootstrap [BOOTSTRAP] and semantic-ui [SEMANTICUI] CSS frameworks for design. The Angular version is being updated periodically and all of the bootstrap components are planned to be replaced by semantic-ui.

In the following sections, these user interfaces will be briefly introduced. In Section 5, these user interfaces will be re-visited while the example demonstration is presented.

3.2.3.1. Single Sign On UIs

As presented in Section 3.2.1.1, C3-Cloud SPS Server implements OpenID Connect Authorization Code Flow [OPENIDCONNECT], i.e. whenever a registered client application initiates authentication flow by making a HTTP Get request to the authorize endpoint of C3-Cloud SPS Server, user is redirected to the **C3-Cloud SPS Manager** for authentication. If the user does not already have an authenticated session, user is redirected to the Single Sign-on UI as depicted in Figure 6 and asked for

his/her credentials. Upon successful sign on, user is redirected to the consent page where he/she is asked to authorize the client to access his user credentials as depicted in Figure 7.

As described in Section 4.1, the C3-Cloud SPS Server will have integration with local Identity Providers of the pilot sites, in this case, SPS Server Authenticate API will redirect the authentication request to the respective authentication server. Even, in this case, for the users which are not already registered to the local Identity Providers of the pilot sites (such as Social Care Workers), C3-Cloud SPS Server handles authentication requests. The situation is similar for SWFT who has opt-out for integration of SPS Server to local Authentication systems, hence for that pilot as well, this SPS Server authentication services will be utilized.

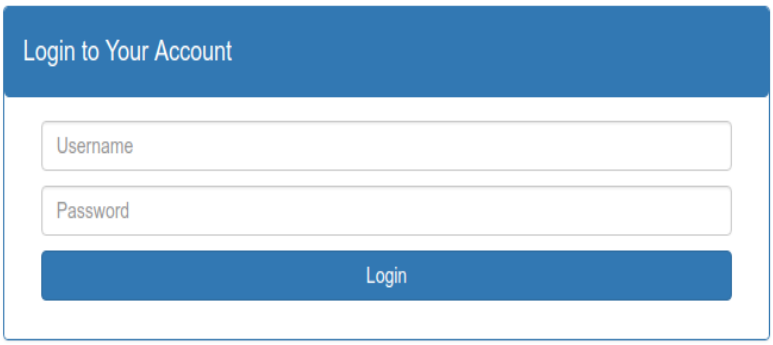

Figure 6 User is redirected to Sign In page

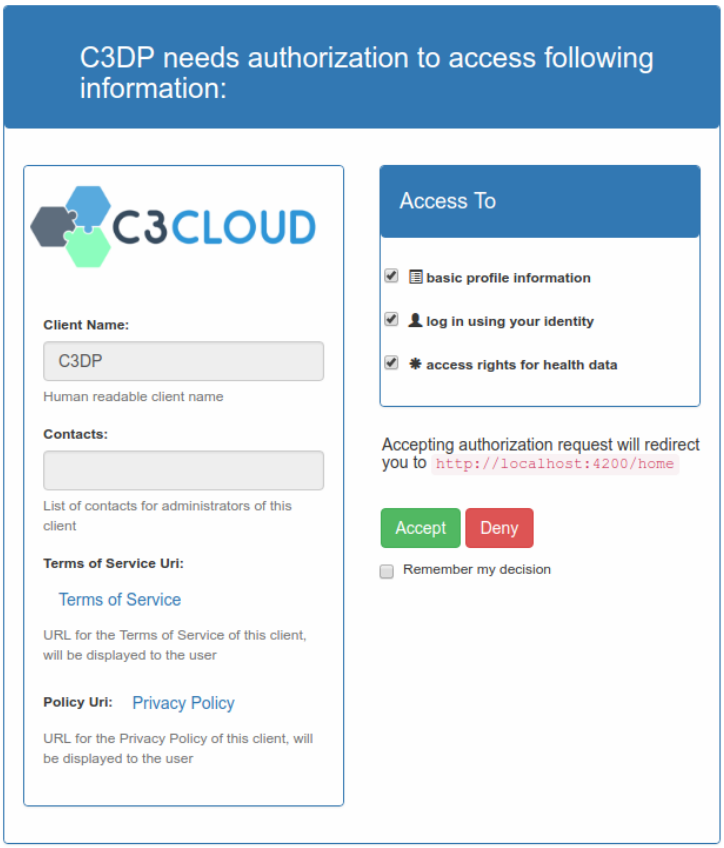

Figure 7 User is asked for Consent

3.2.3.2. Authorization Policy Management UI

In C3-Cloud, privacy policies will be defined per pilot site, by the administrator of the pilot site.

Privacy policy definition consists of set of rules and options that completely identifies the authorization profile of the pilot site. Each pilot site will have a privacy policy that defines the actions that can be carried out by certain roles and resources. Privacy policy definition contains the following fields:

Table 7 Privacy Policy definition

id	Unique identifier of the policy.
name	Human readable name of the policy.
description	Simple description that explains the purpose of policy.
author_id	Identifier of the user who defined the policy.
realm_id	Identifier of the realm that policy belongs to. “C3-Cloud” is defined as a realm.
group_id	Identifier of the group (If the policy defined for the group) that policy belongs to. In C3-Cloud a group id is created per pilot site.
patient_access	Option that are used to decide which set of patients a practitioner can access. Options are; • **realm**: Practitioners can access every patient in their realm • **group**: Practitioners can access every patient in their group • **care-team:** Patients are only accessed by the practitioners within their care-team. In C3-Cloud realm, it is set as group by default.

isBase	Flag that indicates if the policy is a base policy
isActive	Flag that indicates if the policy is the active policy of the author
rules	Rules that defines actions that can be carried out by certain roles and resources.

In C3-Cloud, the following roles have been defined:

Table 8 Draft C3-Cloud roles

practitioner	A Practitioner providing care with support of C3-Cloud
nurse	A Nurse supporting care in C3-Cloud
assistant_nurse	An Assistant Nurse supporting care in C3-Cloud
social_care_worker	A Social Care worker supporting care in C3-Cloud
patient	A Patient in C3-Cloud
Informal care giver	An informal care giver acting on behalf of patient
group_admin	Administrators of pilot sites involved in C3-Cloud

Please note that based on the needs of the pilot sites, further additional roles can be defined easily.

Rules define the access rights of a role on specific protected resource. There is a rule definition in **C3-Cloud SPS Server** for each relation between a role and a resource set. Rule definitions contain following fields:

Table 9 Rule definition

resourceSetId	Indicates which resource is affected by this rule.
roleId	Indicates which role can access this rule.
policyId	ID of the policy that this rule belongs to.
permissions	It defines which permissions are given for resource set (read or write)

The following is the rule definition that identifies the relation between Practitioner and CarePlan resource:

```
{
    "resourceSetId": "CarePlan",
    "roleId": "practitioner",
    “policy_id”: “c3cloud_privacy_policy”
    "permissions": {
        "read": 1,
        "write": 1,
    }
}
```

C3-Cloud SPS Manager provides an interface to define these policies graphically, as presented in Figure 8.

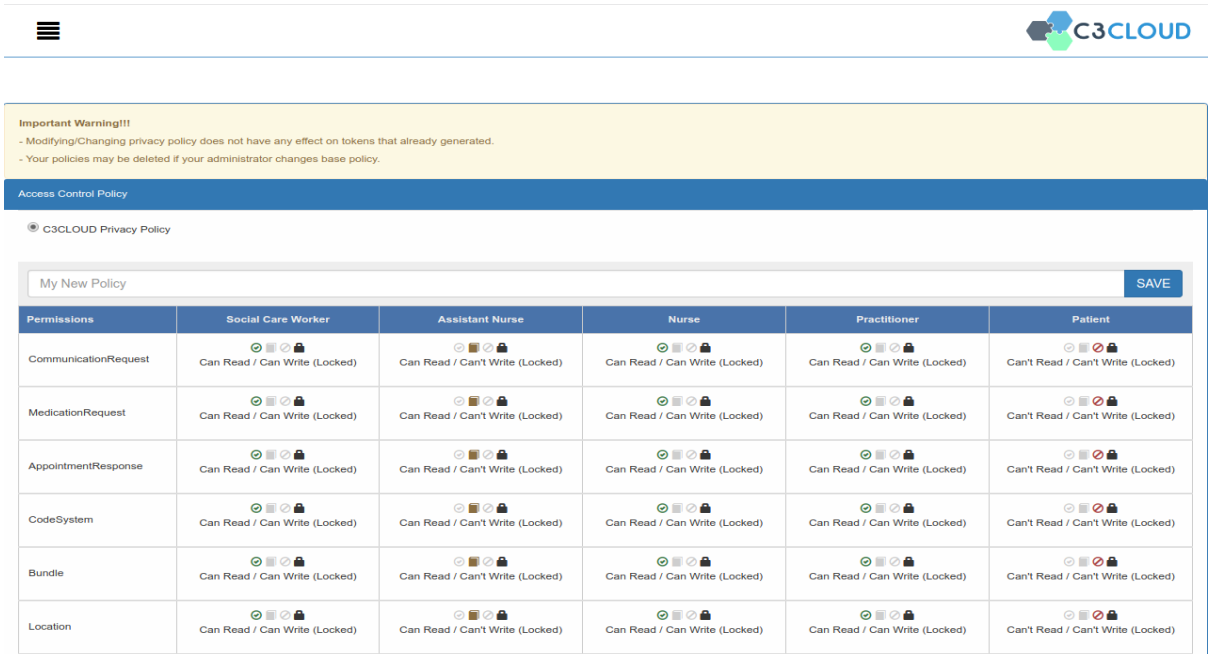

Figure 8 Policy management screen

3.2.3.3. User Registration UI

An administrator with user registration permissions can register new users via the user registration form. The permissions that the newly registered users will receive depends on the role settings at the time of registration. Figure 9 is an example of practitioner registration performed by an administrator:

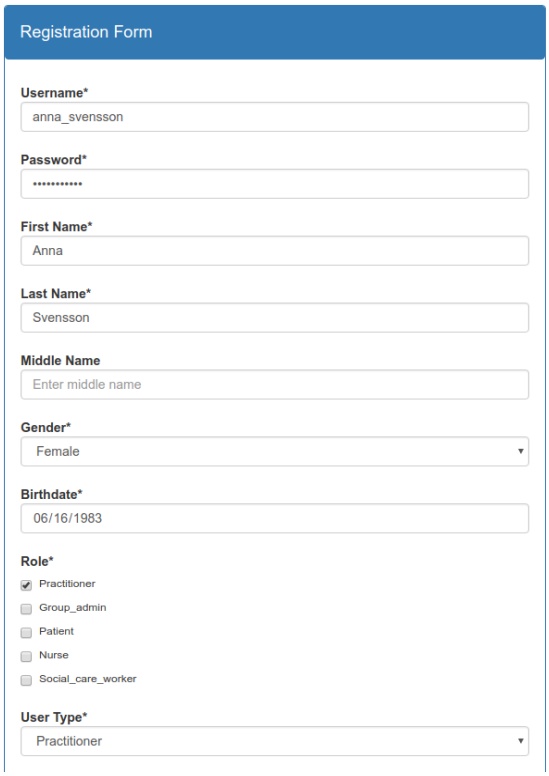

Figure 9 User information entered

3.2.3.4. Client Registration UI

In order to use **OpenID Connect 1.0** endpoints served by the **C3-Cloud SPS Server**, all clients should be registered to **C3-Cloud SPS Server**. Users can register a new client or modify an existing client application using the client registration form (Figure 10):

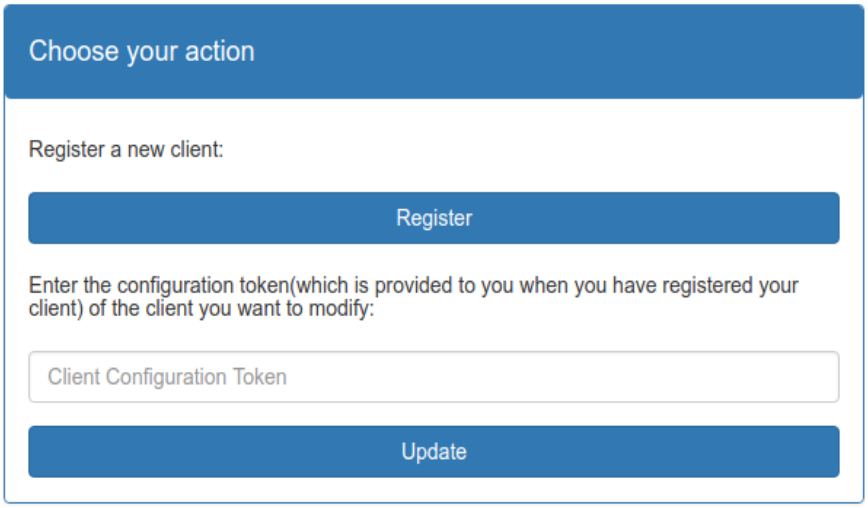

Figure 10 Client registration menu

3.2.3.5. Audit Viewer UI

Audit Viewer UI enables the users with administrator role to see audit logs of all the logged operations in C3-Cloud, such as:

- incoming authentication requests to C3-Cloud SPS;
- queries to C3-Cloud FHIR Repository;

- successful/failed read and write attempts to the resources in C3-Cloud FHIR Repository (such as conditions, lab results, care plans, etc.);
- successful/failed login attempts to C3DP.

See Table 10 for the full list of events that will be logged in C3-Cloud.

The user can filter the logs by date period, operation type (Create, Read, Update, Delete, Execute), outcome status (whether request succeeded or failed) or the related participants of the action, as can be seen in Figure 11.

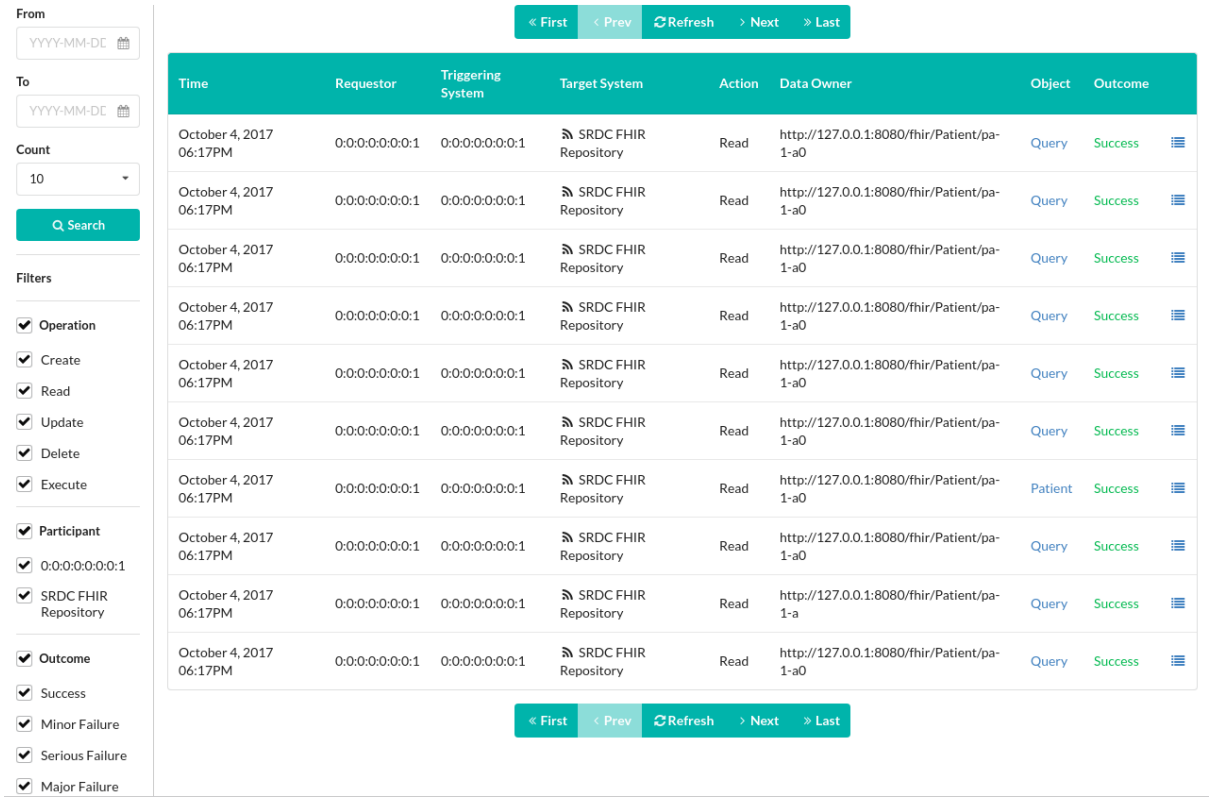

Figure 11 Audit Viewer UI

3.2.4. C3-Cloud Audit Record Repository

In the C3-Cloud architecture, user accountability is ensured through an audit trail mechanism. Through the interfaces provided by the C3-Cloud Audit Module, each C3-Cloud system capable of recording events related to system and personal health information (PHI) processes (i.e. creation, access, modification, as well as import, export or other disclosure of PHI). For each of these events, control information is enabled to be recorded: i.e. time of event, identity and the role of the user, the identity of the subject of care, and the nature of the audited event. The audit trail enables (1) to audit activities, (2) to assess compliance with the domain's policies, (3) to detect instances of non-compliant behavior, and (4) to facilitate detection of improper creation, access, modification and deletion of personal health information.

Audit trails are represented in the FHIR AuditEvent resource [AUDITEVENT], which is based on the IHE Audit Trail and Node Authentication integration profile [ATNA] audit definitions, and exchanged with other C3-Cloud components according to FHIR RESTful API. As Audit Record Repository, the existing C3-Cloud FHIR Repository is being used, the registered clients to C3-Cloud SPS Server, do automatically has access rights to store Audit Event Resources to C3-Cloud FHIR Repository.

C3-Cloud Audit Trail Record information model is based on the FHIR STU3 AuditEvent Resource model [AUDITEVENT]. We have defined our own AuditEvent profile, which is a specialization of the

generic AuditEvent resource for use in the C3-Cloud context. In FHIR, profiling basically involves changing cardinality of attributes, removing some optional attributes and adding some new attributes to existing resources according to the needs of a specific context. An initial version of AuditEvent Profile has already been presented in D3.3 Section 4.4 [D3.3], therefore will not be detailed here. An overview of AuditEvent Resource is depicted in Figure 12:

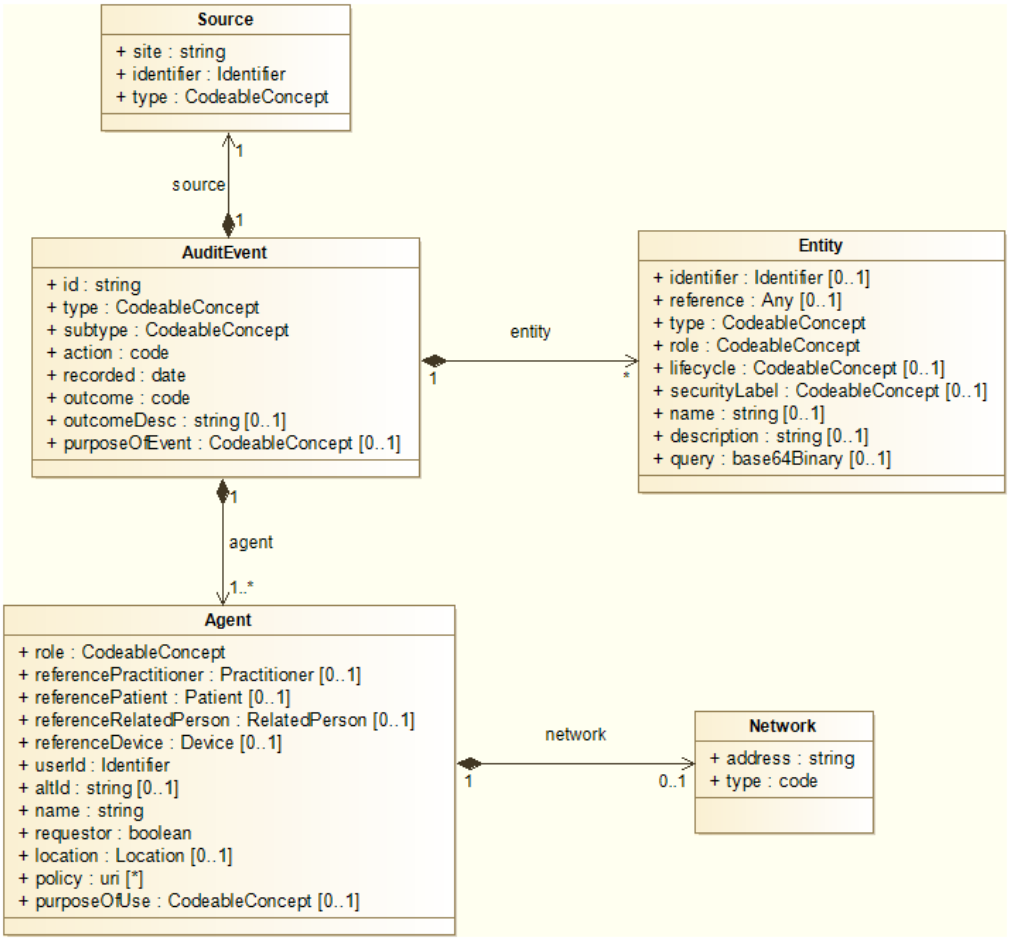

Figure 12 Audit Trail Model [D3.3]

The following table list the events that will be logged (audit events) in C3-Cloud:

Table 10 List of Events that will be logged in C3-Cloud

User Login	Whenever a C3-Cloud user logins into the C3-Cloud components, the event will be logged.
User Logout	Whenever a C3-Cloud user logouts from the C3-Cloud components, the event will be logged.
FHIR Rest Operations	All Restful operations to C3-Cloud FHIR Repository; resource creation, update, search will be logged. These include the updates into & reads from C3-Cloud FHIR Repository by C3DP, TIS and PEP
Authorization Requests	The C3-Cloud SPS Server will log the authorization requests from other users for a specific patient's records.

Technical Interoperability-Data Sync	Whenever TIS triggers a data synchronization with Local Care systems, this event will be logged
Semantic Interoperability - Structural Mapping	Whenever SIS Structural Mapping API is called, this event will be logged
Semantic Interoperability Code Mapping	Whenever SIS code Mapping API is called, this event will be logged
Changes on Privacy Policies	The Authorization and Policy manager will log if the policy definition has been updated

In the FHIR STU3 AuditEvent Resource model, the value sets provided for 'type' and 'subtype' attributes are extensible. The following table maps the events to be audited to values from these values sets, by providing extensions where necessary:

Table 11 'Type' and 'subtype' value sets and its mapping to the types of C3-Cloud audit event types

Events to be Audited	**Selected code for "type" attribute [code (Display), System]**	**Selected code for "subtype" attribute [code (Display), System]**
User Login	110114 (User Authentication), http://dicom.nema.org/resources/ontology/DCM	110122 (Login), http://dicom.nema.org/resources/ontology/DCM
User Logout	110114 (User Authentication), http://dicom.nema.org/resources/ontology/DCM	110123 (Logout), http://dicom.nema.org/resources/ontology/DCM
FHIR Rest Operations	rest (RESTful Operation), http://hl7.org/fhir/audit-event-type	read/vread/update/delete/create/search, http://hl7.org/fhir/restful-interaction
Authorization Requests	110100 (Application Activity), http://dicom.nema.org/resources/ontology/DCM	authorizationRequest, http://www.c3-cloud.eu/fhir/ValueSet/auditevent-subtype
Technical Interoperability-Data Sync	110100 (Application Activity), http://dicom.nema.org/resources/ontology/DCM	dataSync, http://www.c3-cloud.eu/fhir/ValueSet/auditevent-subtype
Semantic Interoperability -Structural Mapping	110100 (Application Activity), http://dicom.nema.org/resources/ontology/DCM	strMapping, http://www.c3-cloud.eu/fhir/ValueSet/auditevent-subtype
Semantic Interoperability Code Mapping	110100 (Application Activity), http://dicom.nema.org/resources/ontology/DCM	codeMapping, http://www.c3-cloud.eu/fhir/ValueSet/auditevent-subtype

Changes on Privacy Policies	110100 (Application Activity), http://dicom.nema.org/resources/ontology/DCM	policyUpdate, http://www.c3-cloud.eu/fhir/ValueSet/auditevent-subtype

In addition to this, in AuditEvent Resource model, while identifying the Actors involved in the event, and also Audit Event Reporter, there is a need to refer to the respective C3-Cloud components. In FHIR AuditEvent Resource this value set is extensible. We have defined the following value set to enable this:

Table 12 C3-Cloud Audit Event Reporter Value Set and its mapping to C3-Cloud components

Component Name	**Identifier (can be mapped to: Agent.userId & Source.identifier)**	**Name (can be mapped to: Agent.name & Source.site)**
C3DP	c3dp.c3-cloud.eu	Coordinated Care and Cure Delivery Platform
C3-Cloud FHIR Repository	fhirrepo.c3-cloud.eu	C3-Cloud FHIR Repository
TIS	tis.c3-cloud.eu	Technical Interoperability Suite
SIS	sis.c3-cloud.eu	Semantic Interoperability Suite
PEP	pep.c3-cloud.eu	Patient Empowerment Platform

4. SECURITY AND PRIVACY SUITE INTEGRATION

4.1. Security and Privacy Suite Integration with Pilot Sites

The progress of SPS integration in each pilot site is presented in the following sub-sections. It should be noted that integration of SPS with the relevant security systems used at the pilot sites is an ongoing work within the scope of WP8 and it is planned to be completed totally by the end of the pilot deployment due Month 30.

4.1.1. End User Authentication

One of the main objectives of Task 6.3 is to enable authentication of the end users into the C3-Cloud applications via their already existing accounts (e.g., username-password, smart card, etc.) provided by the local authorities. Hence, SPS needs to be integrated with the existing Identity Provider (IdP) systems of the pilot sites. No new C3-Cloud specific user account will be created unless there is no integration possibility.

During the pilot application requirements analysis phase, SRDC collected the details of the Identity Provider systems used in the pilot sites. During the conceptual design phase, the requirements from the pilot sites have been taken into account and the following two-fold solution has been proposed:

1. OpenID Connect enabling adapters will be developed on top of the local identity provider system(s) used by the pilot sites for care team member authentication & authorization.
2. OpenID Connect compliant C3-Cloud Identity Provider will be provided as the fallback option, i.e. when it is not possible to integrate with the local systems for some types of users like social care workers.

Within the scope of Task 6.3, in order to discuss detailed integration options and agree on the concrete integration method with each pilot site, SRDC prepared a dedicated "C3-Cloud User Authentication and Authorization" guide and shared with the pilot sites. Two alternative methods of integration have been proposed in this guide, which are also the possible integration options provided by the open source Keycloak Identity and Access Management System [KEYCLOAK]. Keycloak is a very stable and widely used open source product by Red Hat. It is a complete OpenID Connect 1.0 and SAML 2.0 [SAML] compliant IdP itself. Furthermore, the more interesting point in terms of Task 6.3 integration needs is that, it can be integrated with existing user databases (including LDAP and Active Directory) and also other identity provider systems. These options were explained to the pilot sites as follows:

1) User Storage Federation

Keycloak can federate existing external user databases. Out of the box, Keycloak has support to connect to existing LDAP or Active Directory servers. The way it works is that when a user logs in, Keycloak will look into its own internal user store to find the user. If it can't find it there it will iterate over every User Storage provider that was configured until it finds a match. This option is pretty straightforward.

2) Identity Provider Brokering

Keycloak can also authenticate users with existing OpenID Connect 1.0 or SAML 2.0 Identity Providers As in the case of user storage federation, this is a matter of configuring the Identity Provider through the admin console of Keycloak. But this is a more complex configuration.

Below, the integration option that is agreed with each pilot site is presented in separate sub-sections.

4.1.1.1. Region Jämtland Härjedalen

The information security experts of the Region Jämtland Härjedalen (RJH) have opted for the identity provider brokering option. User storage federation option is not possible because direct integration with RJH Active Directory is not allowed. RJH is using Microsoft Active Directory Federation Services (MS AD FS) as their IdP [MS AD FS]. There have been discussions to proceed with integrated via SAML

2.0 based assertions. However, during the summer of 2017, RJH was already planning to update their IdP to the latest release, i.e. MS AD FS 2016, and by Month 18 of the C3-Cloud project, this update is already taking place. According to the specifications, in addition to SAML, MS AD FS 2016 supports OpenID Connect 1.0 directly. As the upgrade is not complete yet, RJH information security experts cannot guarantee whether their installation will have OpenID Connect support or not. It has been agreed that after the upgrade is complete, if OpenID Connect support is working, then the IdP brokering will be achieved via this standard. If not, then IdP brokering via SAML assertions will be attempted.

What has been discussed so far applies to the health professional users whose user accounts are maintained by RJH directly. There will also be social care workers involved in C3-Cloud piloting who are not RJH staff. As it is not possible to integrate with social care workers' IdPs, it has been agreed to use the OpenID Connect 1.0 compliant C3-Cloud fallback IdP that is provided within the SPS for these users. Integration of SPS with the C3DP is already complete in this respect. Their user accounts will be created before the start of the pilot.

4.1.1.2. Osakidetza Basque Country

Detailed technical discussions with the information security experts of Osakidetza revealed that the situation in Osakidetza is a bit different from the other pilot sites. The user accounts and authentication of all Osakidetza workers (e.g., professional care givers, medical specialists) are managed by an LDAP based solution called Norbide. Norbide is hosted in Osakidetza and all systems delegate authentication to it. Osakidetza experts have indicated that neither user storage federation nor IdP brokering is valid for Osakidetza due to following reasons:

- User Storage Federation: LDAP is hosted in the Intranet and is not accessible from the Internet for security reasons.
- Identity Provider Brokering: Norbide doesn't implement standard protocols such as OpenID Connect, OAuth 2.0 or SAML 2.0.

Instead, Osakidetza security experts have made another proposal: JWT Token Bearer. In short, the proposal is based on sharing a signed JWT token. A JWT token is signed with a secret that is shared among all systems. This way, all systems can sign and verify tokens and trust each other. This proprietary solution is already being used by Osakidetza in a number of single sign-on integrations with local web and mobile applications.

The authentication flow is depicted in Figure 13. The same flow is valid for patient authentication but via different systems: Carpeta de Salud instead of Osabide Global and Izenpe instead of Norbide.

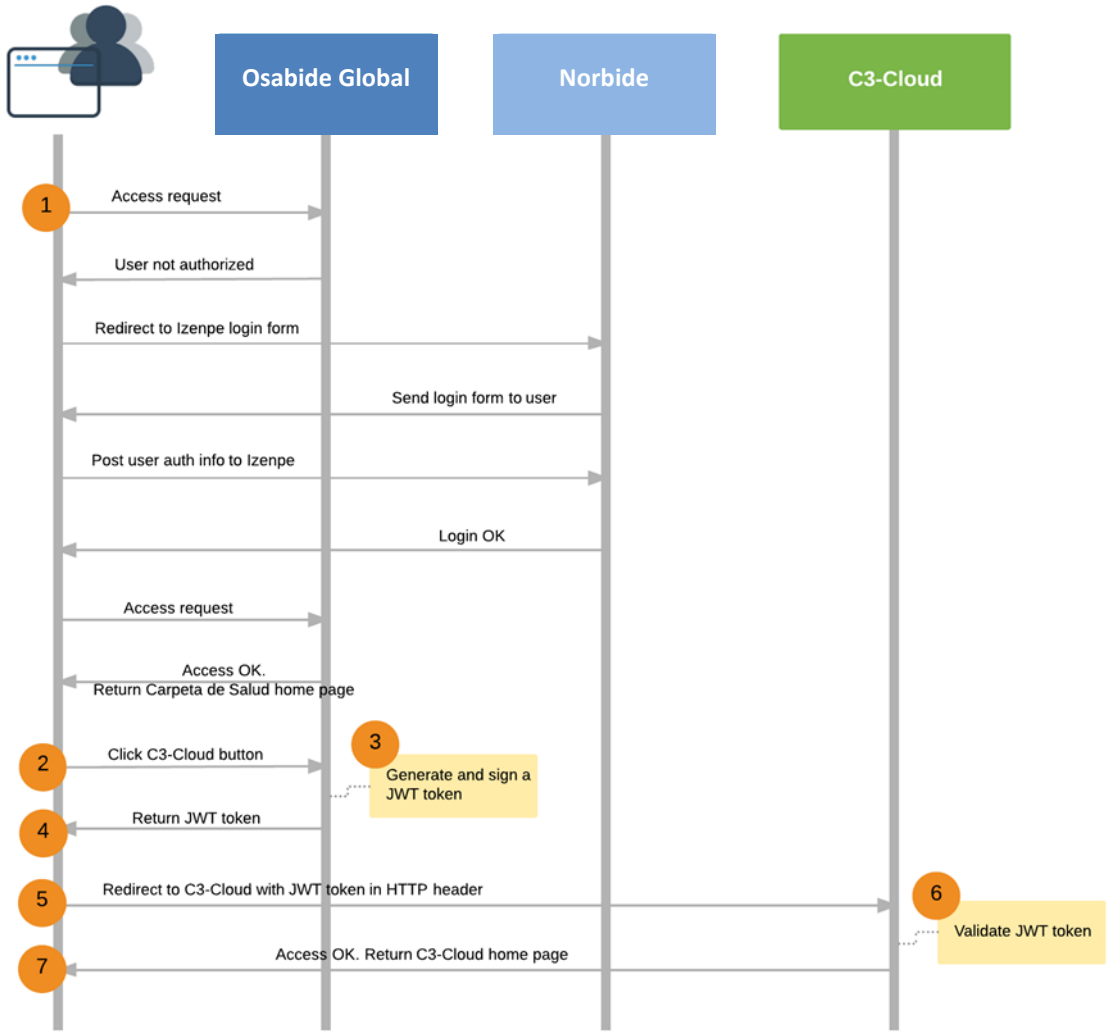

Figure 13 Osabide Global – C3-Cloud Authentication Flow

Another important request by the Osakidetza information security experts is that, no Osakidetza worker will be able to access C3DP Care Plan Management application directly. They will always first login to their EHR system Osabide Global, which they do every day for their operational work. Osabide Global will then provide a link to C3DP, which will direct the user to the C3DP when clicked. Thanks to integration via JWT token sharing as explained above, single sign-on will be achieved and the users will not need to be authenticated again in C3DP. As of Month 18, SRDC and Osakidetza experts are working on finalizing the JWT token structure.

As in the case of RJH, there will also be social care workers involved in C3-Cloud piloting who are not Osakidetza staff. As it is not possible to integrate with social care workers' IdPs, it has been agreed to use the OpenID Connect 1.0 compliant C3-Cloud fallback IdP that is provided within the SPS for these users. Their user accounts will be created before the start of the pilot.

4.1.1.3. South Warwickshire NHS Foundation Trust

The information security experts of the South Warwickshire NHS Foundation Trust (SWFT) clearly indicated that they cannot attempt integration with their IdP for less than 100 users. It is planned to involve 16 professional users from SWFT in the pilot studies, therefore it is not possible to proceed with authentication integration.

SWFT agreed to use the OpenID Connect 1.0 compliant C3-Cloud fallback IdP that is provided within the SPS for all professional users. Integration of SPS with the C3DP is already complete in this respect, hence it can be stated that authentication of SWFT care team members is ready to be handled. The only remaining step is to create user accounts for the actual users before starting the pilot.

4.1.2. Access Control

As explained in the earlier sections, SPS is able to manage access control rules in detail at the level of structural roles and functional roles per each HL7 FHIR resource type. Hence, for example it is possible to restrictor allow access to condition resources of a specific patient for a specific health professional. These capabilities have been explained to the pilot site representatives in various meetings and email exchanges to collect pilot site specific access rules. The experts from the pilot sites expressed that it is not necessary to have too specific access control rules for the purposes of pilot studies, but it is good to know that the technology is already there for use beyond the pilot.

In the end, all pilot sites agreed that functional roles (e.g., GP of a specific patient) will not be used; all health professionals involved in the pilot studies will have access to data of all patients recruited in their pilots. A few pilot sites indicated that they can restrict access of some structural roles (e.g., assistant nurse) for some actions. It has been agreed to define possible actions that can take place via C3DP at the following detail:

a. Read medical data
b. Write medical data (this will be rare but some new medical data might be created via C3DP, e.g. some risk assessments)
c. Read care plan
d. Write care plan

The access control roles provided by each pilot site are presented in the following sub-sections.

4.1.2.1. Region Jämtland Härjedalen

Assistant nurse will have only read access right to medical data and the care plan. All other structural roles will have both read and write access rights to both medical data and care plan.

4.1.2.2. Osakidetza Basque Country

As of Month 18, Osakidetza is still working on finalizing the access control rules per each structural role.

4.1.2.3. South Warwickshire NHS Foundation Trust

All structural roles will have both read and write access to both medical data and care plan.

4.2. Security and Privacy Suite Integration with the Rest of the C3-Cloud Components

As of Month 18, the integration of C3DP with the SPS Server for end-user authentication via the fallback C3-Cloud Identity Provider has been completed. The integration of the SPS Server with the C3-Cloud FHIR Repository has also been completed. The work is ongoing for integration of the C3DP with this secured C3-Cloud FHIR Repository. The integration of SPS with other components like TIS, PEP and SPS has not been initiated yet.

It should be noted that integration of SPS with the other C3-Cloud software components is an ongoing work within the scope of Task 7.4 and it is planned to be completed by Month 24.

4.3. Communication Security

Communication among all integrated C3-Cloud software components and the local systems of the pilot sites will be protected by TLS v1.2 protocol with mutual authentication. All transferring data will be encrypted this way. The implementations will conform to the IHE Audit Trail and Node Authentication (ATNA) specification [ATNA].

5. DESCRIPTION OF THE DEMONSTRATOR

5.1. Demonstration steps

The sample C3-Cloud SPS server is deployed and accessible from: http://app.srdc.com.tr/c3cloud/onauth/onauth-manager/. For demonstration purposes the following access credentials can be used:

user: practitioner2_c3dp

password: password

The demonstration involves creating a new practitioner by an administrator, configuration of users' access control policies and the authentication process of this practitioner to the C3DP application.

5.1.1. Creating Care Team Member Account

The users with administrator role can navigate to user registration page mentioned in Section 3.2.2 and registers a new practitioner by providing the information of this practitioner. User credentials for this practitioner is stored on the authentication server with this information if the registration process succeeded. Figure 14 is an example for registration Anna Svensson as a practitioner.

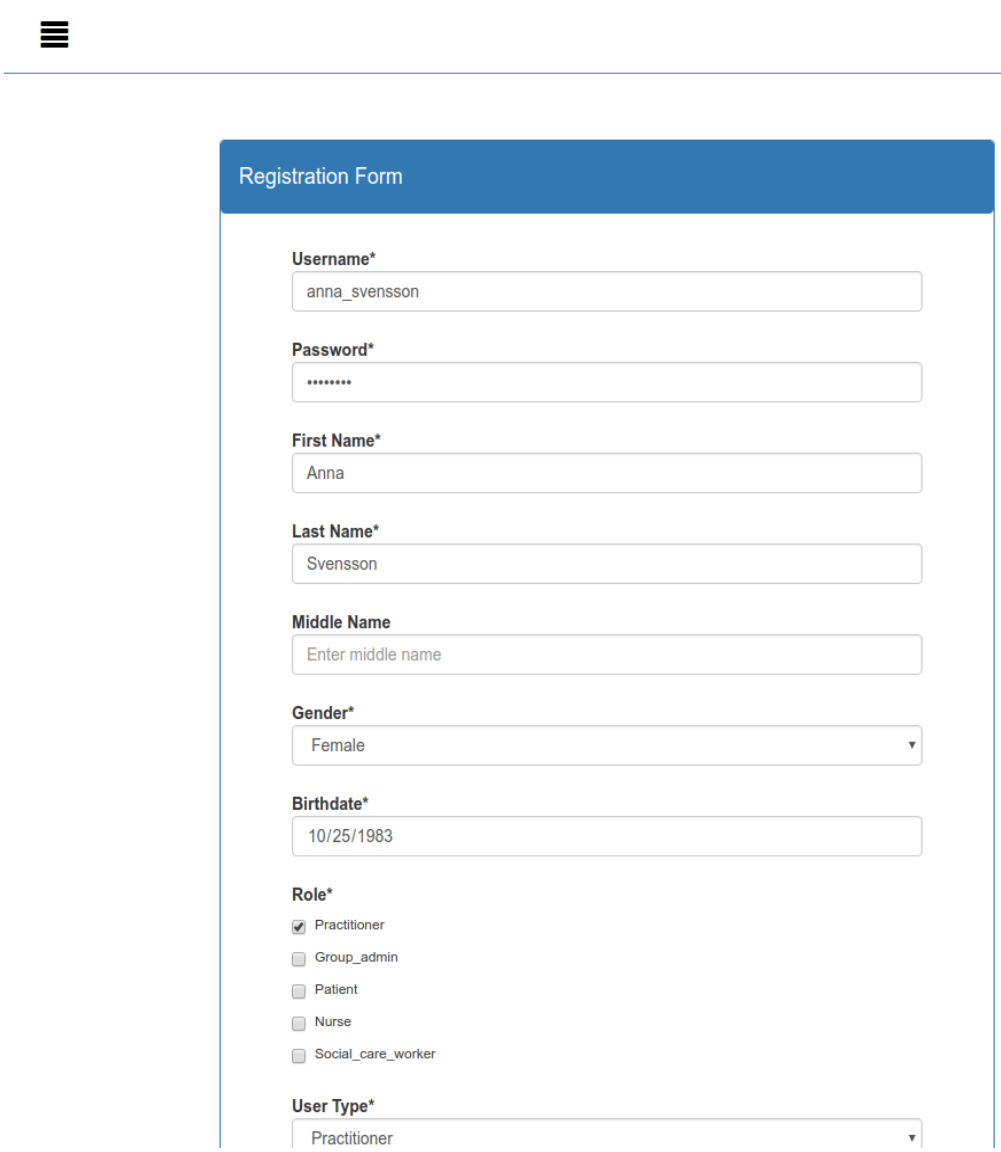

Figure 14 Practitioner Registration

5.1.2. Defining Access Control Policies

The administrator can configure the base access policies for the user roles. Access Policies page allows the administrator to configure the rules that defines which user role has permission to read or update which patient resources as depicted in Figure 15.

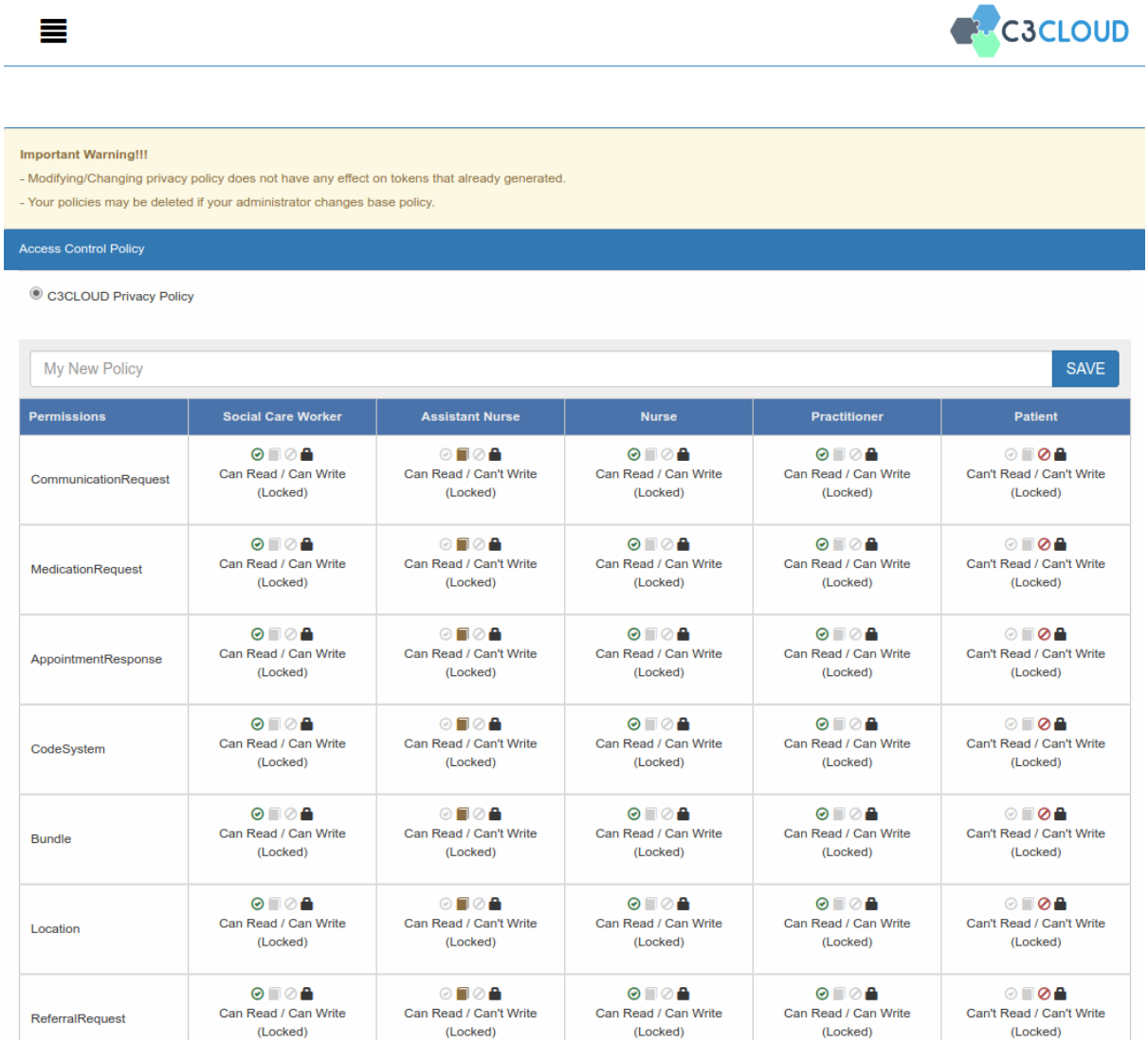

Figure 15 Policy Management

5.1.3. Authentication of Users to Client Application (Single Sign on) & Accessing C3-Cloud FHIR Repository

A practitioner has to be authenticated by the SPS server to use the C3DP application. The examples below are illustrating the authentication of the practitioner Anna Svensson when she using the C3DP application for the first time.

When the practitioner navigates to the C3DP, the application checks if an active session exists for the user. If an active session is not found, C3DP redirects the user to the SPS server for authentication. The details of the HTTP request for the authentication are shown in **Error! Reference source not found.**. C3DP sends the request to the authentication API (Figure 16-1) by setting the response type as code and redirect URI as C3DP home page. Authentication endpoint transfers the request to the login page (Figure 16-2).

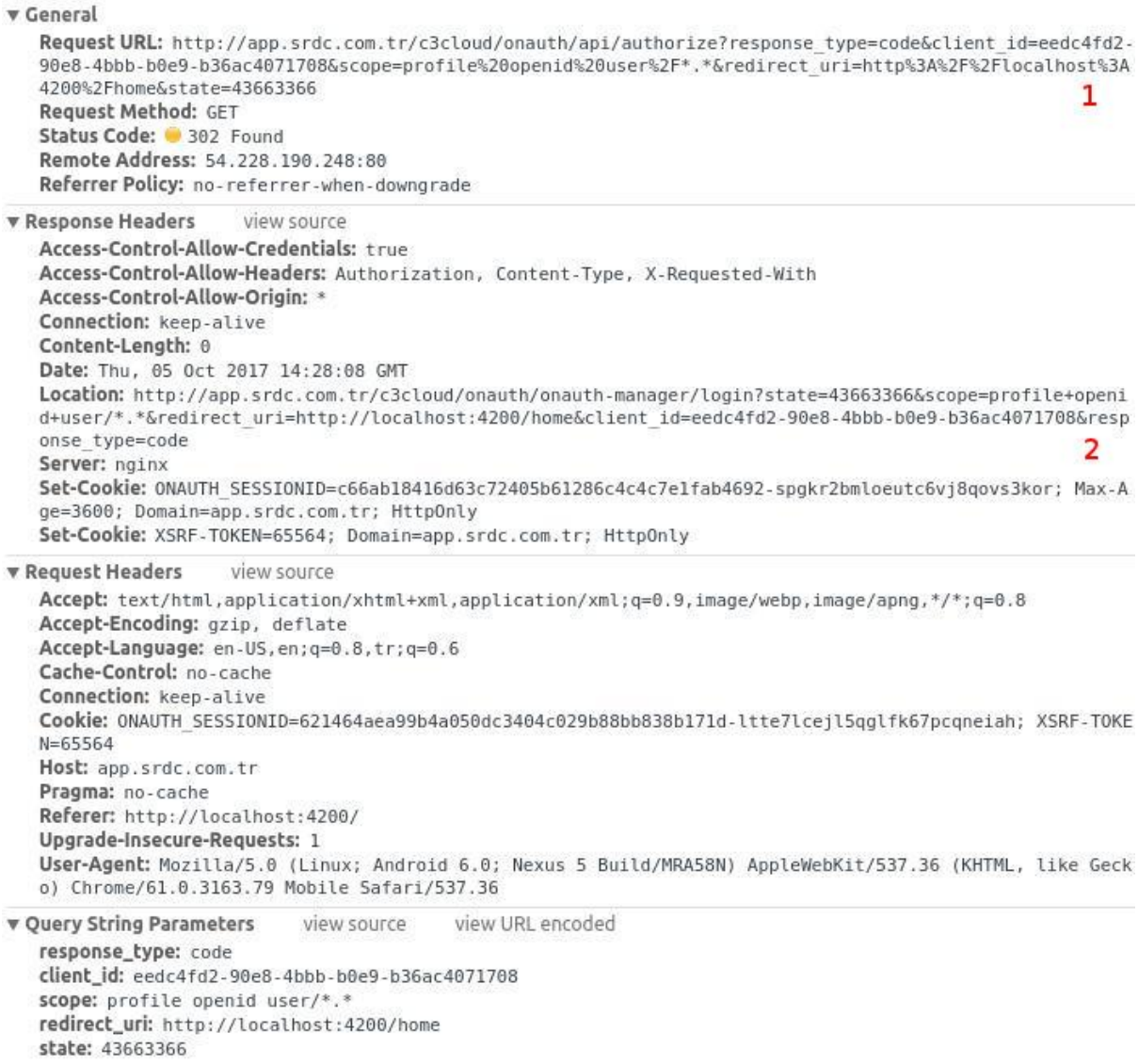

Figure 16 Request to Authentication Service

After these processes handled in the background, the user can log-in via the login page with the credentials, which has been provided by the system administrator as depicted in Figure 17.

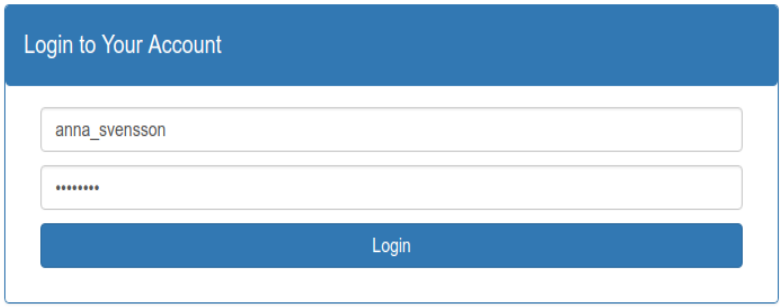

Figure 17 Anna Svensson Login

If it is the first time Anna Svensson is using the application or the "Remember my decision" option is not checked before, she will be asked for consent (Figure 18).

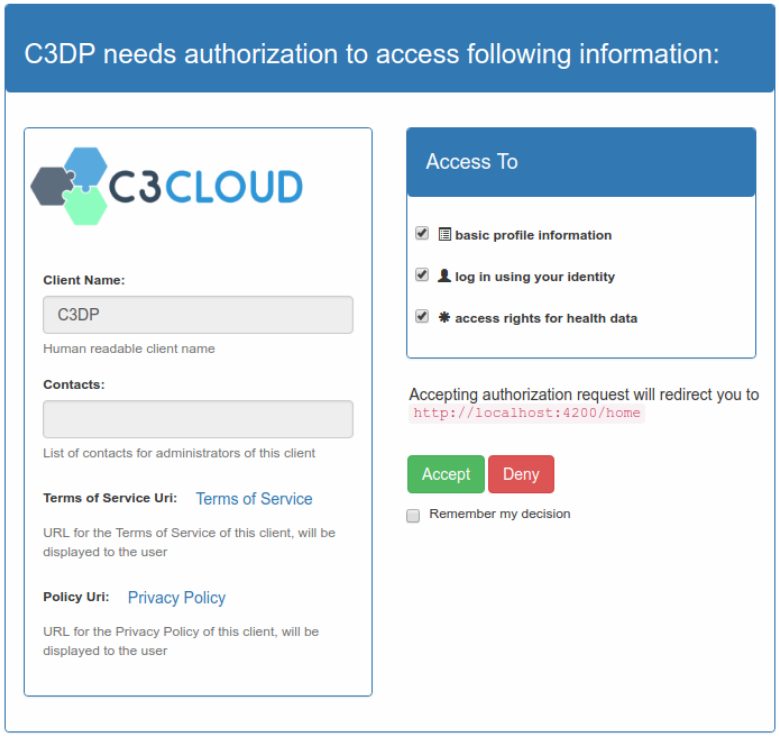

Figure 18 Anna Svensson gives her consent

After accepting the consent, Anna Svensson will be redirected to the C3DP with a code and state information for the new session. Figure 19 Figure 20 shows the request details (code and state parameters)

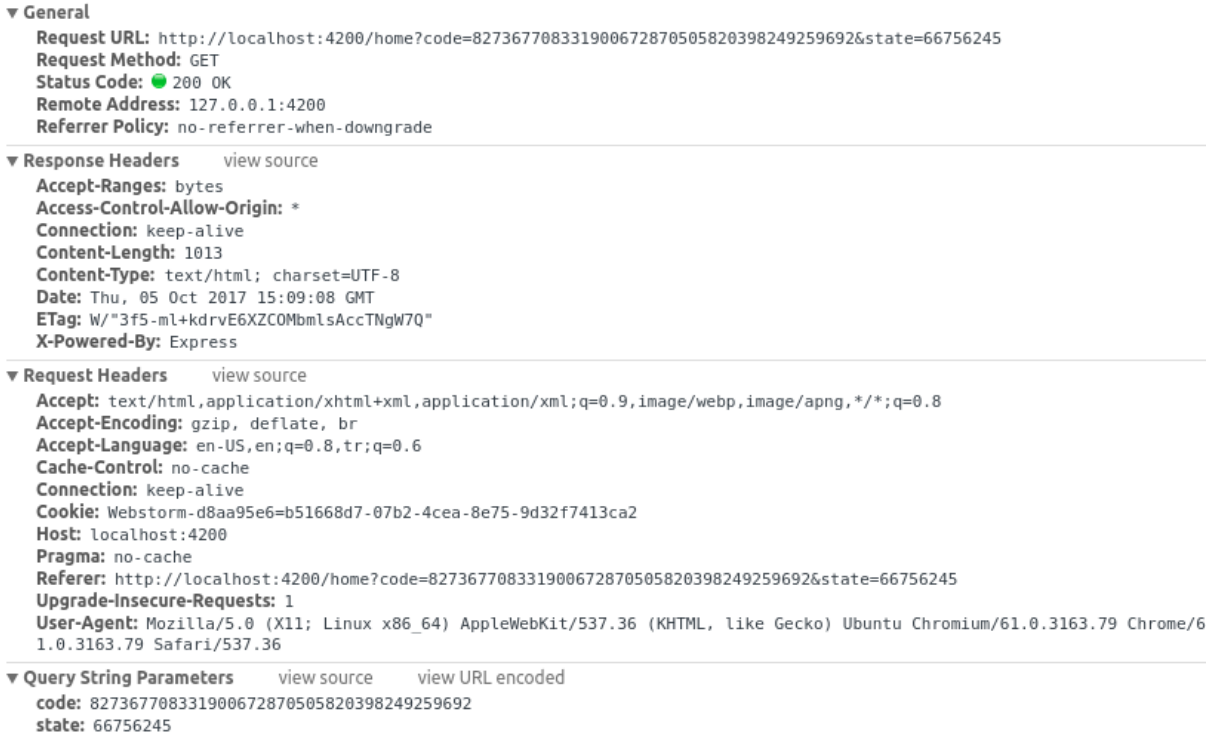

Figure 19 Redirection to C3DP by Authentication Service

When C3DP gets the code and state, it will request for an access token with this information. The structure of the request that is sent by C3DP to the token endpoint and the received token from the server are shown in the figures below (Figure 20, Figure 21).

▼ General
Request URL: http://app.srdc.com.tr/c3cloud/onauth/api/token
Request Method: POST
Status Code: 200 OK
Remote Address: 54.228.190.248:80
Referrer Policy: no-referrer-when-downgrade

▼ Response Headers view source
Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: Authorization, Content-Type, X-Requested-With
Access-Control-Allow-Origin: *
Cache-Control: no-store
Connection: keep-alive
Content-Length: 3716
Content-Type: application/json
Date: Thu, 05 Oct 2017 15:08:55 GMT
Pragma: no-cache
Server: nginx

▼ Request Headers view source
Accept: application/json, text/plain, */*
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8,tr;q=0.6
Cache-Control: no-cache
Connection: keep-alive
Content-Length: 161
Content-Type: application/x-www-form-urlencoded
Host: app.srdc.com.tr
Origin: http://localhost:4200
Pragma: no-cache
Referer: http://localhost:4200/home?code=827367708331900672870505820398249259692&state=66756245
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/61.0.3163.79 Chrome/
1.0.3163.79 Safari/537.36

▼ Form Data view source view URL encoded
grant_type: authorization_code
client_id: eedc4fd2-90e8-4bbb-b0e9-b36ac4071708
redirect_uri: http://localhost:4200/home
code: 827367708331900672870505820398249259692

Figure 20 Access Token Request Message

▼ {access_token: "ib9a89a39g12c492fgja1haijhi0i1", token_type: "Bearer",…}
access_token: "ib9a89a39g12c492fgja1haijhi0i1"
expires_in: 3600
id_token: "eyJraWQiOiJlYjAwZDg3Yy1mZTZhLTQ4MjctYTQ1Mi1jM2YxMTkyN2M1NTciLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJjNjQ1NjdhZi1hZWJ
scope: "user/VisionPrescription.write user/TestReport.write user/DeviceMetric.write user/Sequence.write user/GraphDefin
token_type: "Bearer"

Figure 21 Access Token received

After getting an access token, Anna Svensson can use C3DP and access the data of the patients using this token (Figure 22).

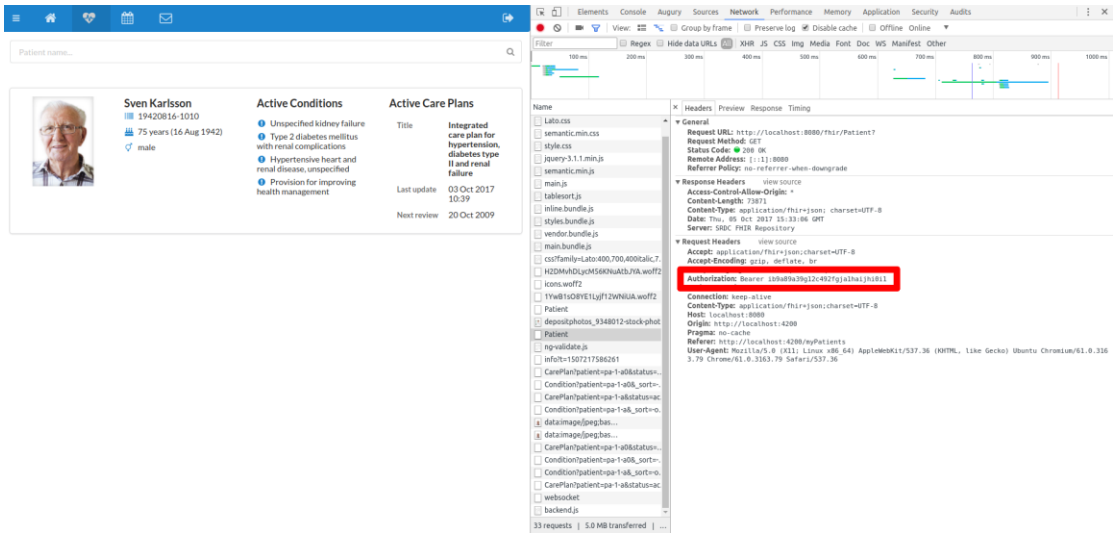

Figure 22 Anna Svensson sees patient details in C3DP

5.1.4. Viewing Audit Records

The users that has administrator permissions can display the audit logs via Access Logs page of the SPS Manager. In the previous section, Anna Svensson searched for her patients and viewed their information. The audit records for these operations are shown in Figure 23.

« First ‹ Prev Refresh › Next » Last

Time	Requestor	Triggering System	Target System	Action	Data Owner	Object	Outcome
October 6, 2017 09:31AM	0:0:0:0:0:0:0:1	0:0:0:0:0:0:0:1	SRDC FHIR Repository	Read	http://127.0.0.1:8080/fhir/Patient/pa-1-a	Query	Success
October 6, 2017 09:31AM	0:0:0:0:0:0:0:1	0:0:0:0:0:0:0:1	SRDC FHIR Repository	Search	http://127.0.0.1:8080/fhir/Patient?	Query	Success

Figure 23 Audit Records for searching patient records from FHIR Repository

6. FUTURE PLANS

As explained and demonstrated in the earlier sections, the centralized implementation of the C3-Cloud Security and Privacy Suite is completed in the timeframe of Task 6.3. Some integration work has also been completed as presented in Section 4. Concise plans are in place for proceeding with the remaining integration activities, both with the other C3-Cloud software components and with the pilot site systems. These integration activities will take place within the scope of Task 7.4 - Development of Coordinated Care and Cure Delivery Platform through Integration of C3-Cloud Components and Task 8.3 - Deployment and Operation of C3-Cloud Pilot Application respectively.

The planned integration work to take place in the following months is summarized below:

- Integration with C3-Cloud software components
    - C3DP integration with the secured C3-Cloud FHIR Repository via SPS
    - PEP integration with the secured C3-Cloud FHIR Repository via SPS
    - TIS integration with the secured C3-Cloud FHIR Repository via SPS
- Integration with pilot sites
    - Region Jämtland Härjedalen
        - Proceed with IdP brokering integration after MS AD FS 2016 upgrade
        - Create user accounts for non-RJH workers to take place in the pilot studies via C3-Cloud fallback IdP
    - Osakidetza Basque Country
        - Proceed with JWT token bearer integration between Osabide Global and C3-Cloud SPS
        - Provide final access control rules for structural roles
        - Create user accounts for non-Osakidetza workers to take place in the pilot studies via C3-Cloud fallback IdP
    - South Warwickshire NHS Foundation Trust
        - Create user accounts for all SWFT care team members to take place in the pilot studies via C3-Cloud fallback IdP
- Common integration
    - Provision and use of necessary digital security certificates for Secure Node authentication (i.e. encryption during data transfer) according to IHE ATNA profile

7. REFERENCES

[ANGULAR] Angular Framework, https://angular.io/

[ATNA] IHE Audit Trail and Node Authentication integration profile, http://wiki.ihe.net/index.php/Audit_Trail_and_Node_Authentication

[AUDITEVENT] FHIR STU3 AuditEvent Resource, http://hl7.org/fhir/auditevent.html

[BOOTSTRAP] Bootstrap CSS Framework, http://getbootstrap.com/

[D3.2] C3-Cloud Deliverable 3.2 - Requirements Specification of the C3-Cloud Architecture

[D3.3] C3-Cloud Deliverable 3.3 – Conceptual Design of the C3-Cloud Architecture

[KEYCLOAK] Keycloak Identity and Access Management System, https://www.keycloak.org

[MS AD FS] Microsoft Active Directory Federation Services, https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/overview/ad-fs-scenarios-for-developers

[OAUTH] OAuth 2.0, https://oauth.net/2/

[OPENIDCONNECT] OpenID Connect Core 1.0, http://openid.net/specs/openid-connect-core-1_0.html

[OPENID-CLIENTREG] OpenID Connect Dynamic Client Registration 1.0, http://openid.net/specs/openid-connect-registration-1_0.html

[SAML] Security Assertion Markup Language (SAML) V2.0, http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html

[SEMANTICUI] Semantic-UI CSS Framework, https://semantic-ui.com/

[TLS] The Transport Layer Security (TLS) Protocol Version 1.2, https://www.ietf.org/rfc/rfc5246.txt