European Security in Health Data Exchange

Deliverable D5.8

Consent Management

Editor(s):	Graham Davidson
Responsible Partner:	Symphonic
Status-Version:	Final - v1.0
Date:	03/07/2018
Distribution level (CO, PU):	CO

Project Number:	GA 727301
Project Title:	SHIELD

Title of Deliverable:	Consent Management
Due Date of Delivery to the EC:	30/06/2018

Workpackage responsible for the Deliverable:	WP5 - Data protection and privacy
Editor(s):	Symphonic
Contributor(s):	Symphonic
Reviewer(s):	Symphonic, Tecnalia (Xabier)
Approved by:	All Partners
Recommended/mandatory readers:	WP2, WP5, WP6

Abstract:	This second version will be the first implementation of the tools and API's aiming at providing the necessary support to WP6 for the initial evaluation of the architecture and functionalities.
Keyword List:	Data Privacy, Masking
Disclaimer	This document reflects only the author's views and neither Agency nor the Commission are responsible for any use that may be made of the information contained therein

Document Description

Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
v0.1	24/04/2018	First draft	Graham Davidson, Derick James
V1.0	03/07/2018	Final	Xabier Larrucea

Table of Contents

List of Figures

Terms and abbreviations

DB	Database
EC	European Commission
GDPR	General Data Protection Regulation
PAP	Policy Administration Point
PDP	Policy Decision Point
PIP	Policy Information Point
UI	User Interface

Executive Summary

This is the second deliverable related to Task 5.3: Consent Management and aims to provide support for initial evaluation of the architecture and functionality. This deliverable has two parts: a prototype and this document which reports on the main progress made in the prototype and provides instructions for deployment and use of the component.

1 Introduction

1.1 About this deliverable

This is the second deliverable related to Task 5.3: Consent Management and aims to provide support for initial evaluation of the architecture and functionality. Symphonic will provide an integrated system to manage and enforce patient consent preferences. A Symphonic decision engine and administration point will allow authorization policies to be defined and evaluated giving greater flexibility than traditional authorization approaches. The system will be comprised of the components described below and highlighted green in figure 1.

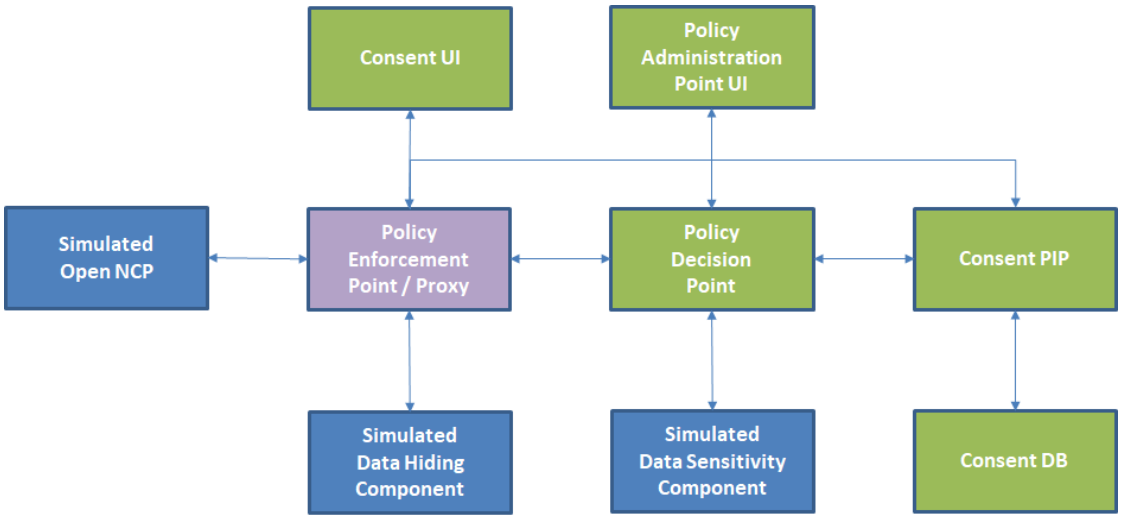

Figure 1: Consent Management Block Diagram

A consent UI and database will facilitate the input and storage of patient consents at a fine-grained level. Patients will have the ability to specify which data to share, the geographic location in which it can be shared and the duration for which the consent should remain active. A consent Policy Information Point (PIP) will make the consent data available to the Policy Decision Point (PDP) where it will form part of the input data upon which authorization decisions will be made. Additional input data regarding hospital and clinician authentication/authorization will be obtained from OpenNCP. The Policy Administration Point provides a UI by which the various authorisation rules used in the PDP can be viewed, input, edited and managed. Based on the input data and authorisation rules defined through the PAP the PDP will return a **permit** or **deny** decision as to whether each particular information request is authorised. In the case of a permit the PDP may also return an obligation to redact the data if necessary.

1.2 Progress made

Having built upon the work done in previous deliverables including Deliverable D5.3 – Consent Management the following components have been developed in prototype format:

1. Symphonic Policy Decision Point (PDP).
2. Symphonic Policy Administration Point (PAP).
3. Consent Management User Interface (UI).
4. Consent Management Database schema.

5. Consent Management Application Programming Interface (API).
6. Consent Management Policy Information Point (PIP).

1.3 Document structure

In the following sections this document will provide Instructions on how to deploy and operate the consent component. Section 2 provides information on how to install / deploy the component. Section 3 gives a user guide for the PAP and PDP. Section 4 is a user guide for the consent.

2 Delivery and Installation

2.1 Package information

For convenience and flexibility, the sub components of the Consent Management Component shown in green in figure 1 are made available as deployable FAT JARs:

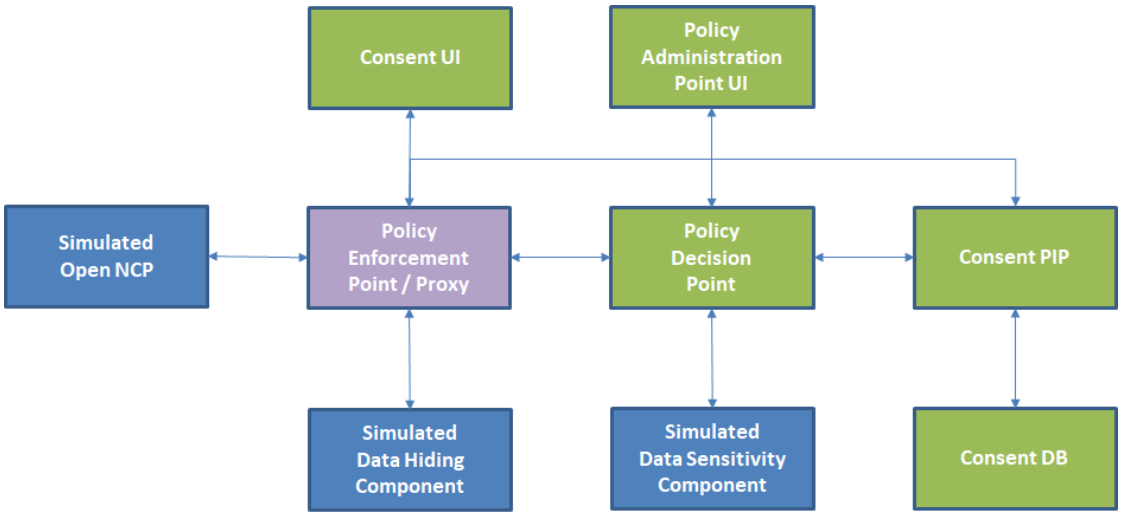

Figure 2: Consent Management Architecture

1. Policy Decision Point (PDP) and Policy Administration Point (PAP) are available as a single JAR which can be copied on to the target machine and executed directly.
2. Consent User Interface (UI), Consent Database (DB) and Consent Policy Information Point (PIP) are available as a single JAR which can be copied on to the target machine and executed directly.

We also support deploying the above JARs using Docker / Docker Compose.

2.2 Installation instructions

This section describes the process of Installing and running the prototype.

2.2.1 PDP / PAP JAR

To install the JAR file, simply copy it to the server. Then run the following command:

```
java -jar symphonic-admin-point-application-<version>-
capsule.jar server configuration.yml
```

This will run the administration point using an embedded H2 database. An embedded UI will be served on http://localhost:8080/, with metrics and health checks available at http://localhost:8081/. The REST API will be available at http://localhost:8080/api/, with swagger documentation available through the UI. Log files will be generated in the working directory.

2.2.2 Consent UI / PIP / DB JAR

To install the JAR file, simply copy it to the server. Then run the following command:

```
java -jar consent.jar
```

This will run the consent management PIP and UI using an embedded H2 database. An embedded UI will be served on http://localhost:8091/, the REST API will be available at http://localhost:8090/api/, with swagger documentation available through the UI. Log files will be generated in the working directory.

2.2.3 Docker

To run the Consent Component using docker-compose, open a command shell and change directory to the distribution directory. There should be a docker-compose.yml file here. To run the PAP, PDP, Consent UI, PIP and DB with docker, run:

```
docker-compose up
```

This will two containers, one each for the PAP / PDP and one for the Consent UI / PIP / DB. The PAP UI is available at http://localhost:8080/, the PAP REST API at http://localhost:8020/api/ http://localhost:8091/, the REST API will be available at http://localhost:8090/api/.

3 PAP / PDP User Guide

The Policy Manager provides the tools for implementing fine-grained and dynamic access control policies, allowing you to govern how the resources (services and data) of your organisation may be used.

You will use the Policy Manager to create policies that answer the question: "should this resource access request be permitted or denied?". In a traditional RBAC (role-based access control) system, this question can be rephrased as "who is the user making the access request, and has the user been assigned one of the roles that are permitted access to this resource?". Certainly, it is possible to model such a policy in Symphonic; however Symphonic is essentially an ABAC (attribute based access control) system, and in such a system the question may be rephrased as "given the facts I know about the user, the resource being accessed, what the user wants to do with the resource, how sure I am the user is who she says she is, and any other pertinent facts about the world at this point in time, should the user's access request be permitted, and is there anything else that must be done in addition to permitting or denying access?". That's quite a mouthful, and this speaks to the inherent power of Symphonic. Fortunately, the Policy Manager makes harnessing this power quite straightforward.

This guide will introduce the features of the Symphonic Policy Manager, and show you how to create access control policies that reflect your business requirements. We'll take a tour of the various concepts involved in policy modelling in Symphonic.

3.1 Policy Sets, Policies and Rules Overview

A typical large organisation may have many hundreds or thousands of conditions and constraints around access control; these are the business rules that define the circumstances under which certain resources may be accessed. Fortunately, these rules can be naturally grouped together, which allows us to reason about them without having to hold them in our head all at once.

For example, there may be a set of policies around authentication, that require a user to have authenticated to a certain level before she many access some type of resource. Another set of policies may gather together all the business rules around accessing the resources of a particular business unit. Yet another set of policies may define audit processes that must be triggered whenever access to some set of restricted resources is attempted.

This structure that is inherent in the problem domain of resource access control is reflected in Symphonic in three types of entities - Policy Sets, Policies, and Rules - and the relationship between them. In this section, we'll take a tour through these three types of entity, see how they are composed together, and take a look at their properties.

3.2 Policies / PolicySets

Navigate to policy manager by clicking the ***Policies*** button on the main navigation bar on the top of the screen.

Figure 3: Navigation

Existing policy nodes will be listed in the navigation panel of the left hand side, organised in a tree structure. It is good practice to add a root PolicySet to hold all other PolicySets, this will be useful later down the line when building a Deployment Package from the entire Policy tree.

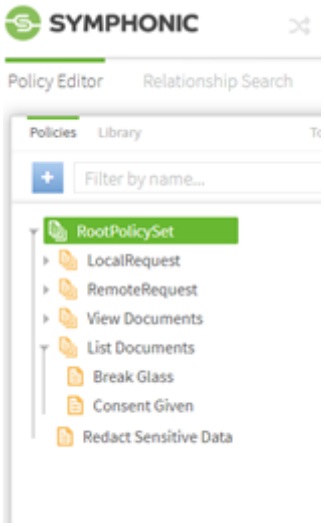

Figure 4: Policy Tree

3.3 Creating Policies / PolicySets

Make sure that the Policies tab is selected on the policy navigation panel and select the blue creation button on from the panel. A popup will appear allowing you to select and start the creation process of either a Policy or PolicySet.

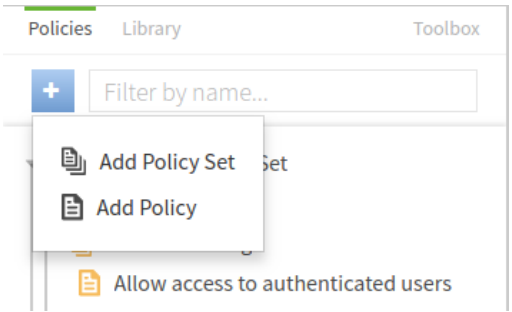

Figure 5: Adding Policies/ PolicySets

Policies and PolicySets can be named anything you like but it is recommended, especially as the Policy tree gets larger and more complex that they are given relevant and contextual names.

When naming Policies you should consider the business rule that they are trying to model and check if they adequately represent the operational policies of the organisation.

Figure 6: Basic Policy

A Policy given the name 'My Basic Policy' used for demonstration in this documentation. Notice the red dot in the top right corner, this signifies that after we have changed the name there are unsaved changes in the Policy. If you try to navigate away from this page you will be presented with a popup that will remind you to save your changes or you can optionally decide to discard these changes.

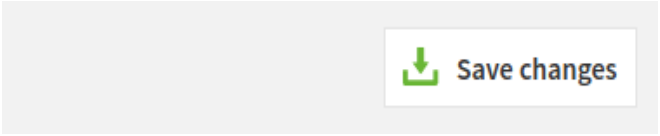

Figure 7: Save Button

To save a Policy you can click the Save Changes button at the bottom of the screen

3.4 Adding targets to the Policy

So far the only change we have made has been to give the Policy a name. Let's add some targets to the Policy so we know which requests this policy applies to.

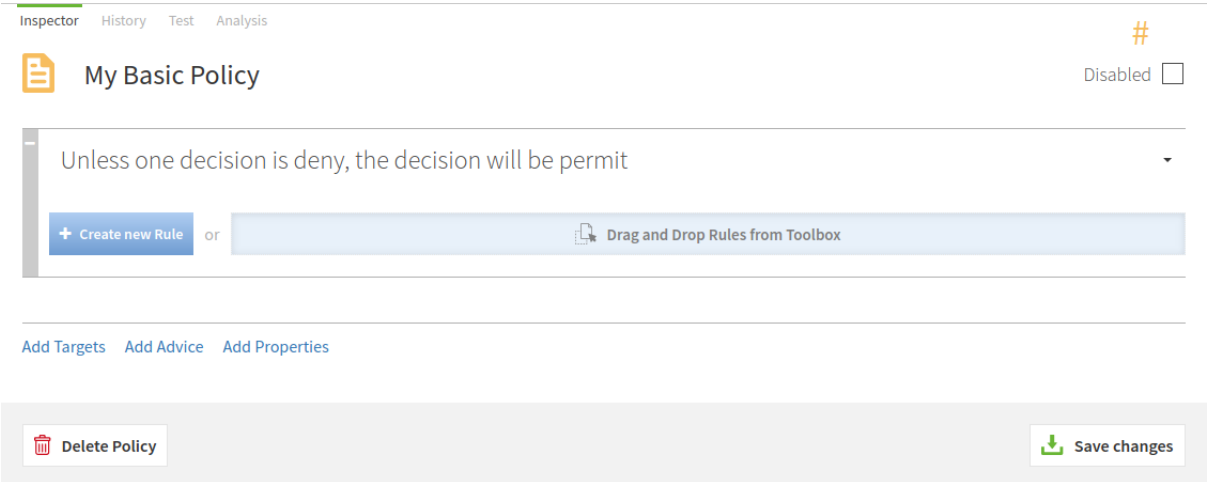

Figure 8: Add Targets Button

Start by clicking the **Add Targets** button which will expand the target section in the Policy

Figure 9: Add Targets Dialog

Notice that this Policy applies to all requests, when a Policy does not have any targets attached then this will be the case. Think of a target as a way of defining a set of access requests that this Policy applies to. If you wanted a Policy to only apply for all requests to a certain database for example then you would add the database domain as a target.

The target section is used in conjunction with the toolbox in the left panel. **Domains**, **Services**, **Identity Classes** and **Actions** can be dragged and dropped into the target section on the Policy. Here you can see the elements created in the Trust Framework. If we wanted to target any Internal requests then that service can be dragged in.

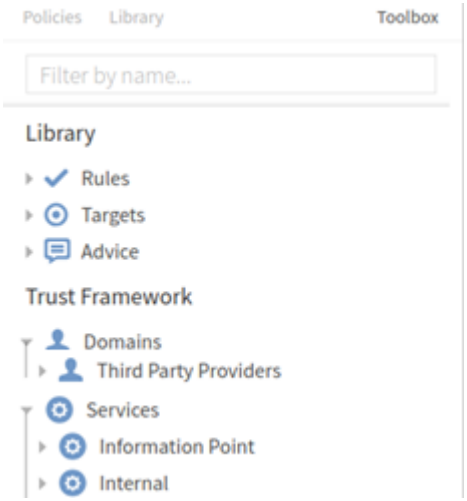

Figure 10: Target Selection

Targets are described in a nested structure. By dragging in a parent element as a target you are also targeting all of its children. By doing this you are actually adding multiple targets as the top level is a target as well.

3.5 Advice

Advice is what is sent back to the governance engine so that the correct action can be taken depending on the response that was evaluated from the Policy itself. If a policy was setup to check the authentication level of a user and the policy evaluated that the user did not have the right access privileges then information could be sent to give some information as to why access was denied.

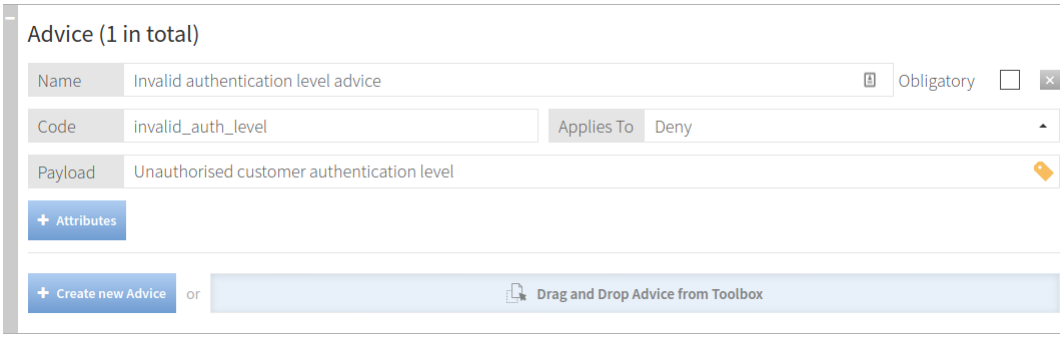

Figure 11: Advice

3.6 Properties

Properties are used to add metadata to a policy in the format of a key value pair.

3.7 Rules and Combining Algorithms

Each policy may have a number of rules which can produce a **permit** or **deny**. To evaluate the overall decision of the Policy, a combining algorithm is applied. The default algorithm set on a new policy is **Unless One Decision is Deny, the Decision will be Permit**. This algorithm will always evaluate to permit unless the decision is deny. A full description of the available algorithms is given at the end of this section.

3.8 Rule Structure

Rules contain conditions, which contain the logic of the that will evaluate to true or false. Each Rule can be given an effect which will be either permit or deny. It is important to note that the effect is what the rule evaluates to when it's child condition or group of conditions evaluates to true. A rule could be set so that if a condition evaluates to true and the effect is set to deny, then that rule will evaluate to deny.

Like Policies and PolicySets, Rules can also have targets which work in the same way. You can use the targets to use a more fine grained approach, for example one rule could target the internal and the external requests.

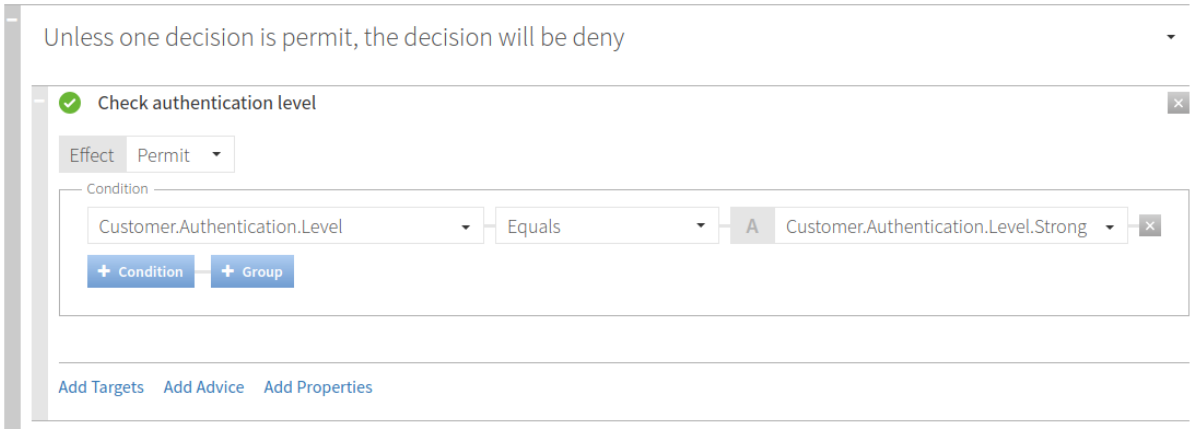

Figure 12: Conditions

If this condition evaluates to true then the effect will be permit.

3.9 Combining Strategies - Truth Tables

	Permit	Deny	NotApplicable	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Permit	Permit	Deny	Permit	Indeterminate (D)	Permit	Indeterminate (DP)
Deny	Deny	Deny	Deny	Deny	Deny	Deny
NotApplicable	Permit	Deny	NotApplicable	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Indeterminate (D)	Indeterminate (D)	Deny	Indeterminate (D)	Indeterminate (D)	Indeterminate	Indeterminate (DP)
Indeterminate (P)	Permit	Deny	Indeterminate (P)	Indeterminate (DP)	Indeterminate (P)	Indeterminate (DP)
Indeterminate (DP)	Indeterminate (DP)	Deny	Indeterminate (DP)	Indeterminate (DP)	Indeterminate (DP)	Indeterminate (DP)

a .Deny-overrides: if any rule evaluates to a deny then it will override any other rule's decision.

	Permit	Deny	NotApplicable	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Permit	Permit	Permit	Permit	Permit	Permit	Permit
Deny	Permit	Deny	Deny	Deny	Indeterminate (P)	Indeterminate (DP)
NotApplicable	Permit	Deny	NotApplicable	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Indeterminate(D)	Permit	Deny	Indeterminate (D)	Indeterminate (D)	Indeterminate (DP)	Indeterminate (DP)
Indeterminate (P)	Permit	Indeterminate (P)	Indeterminate (P)	Indeterminate (DP)	Indeterminate (P)	Indeterminate (DP)
Indeterminate (DP)	Permit	Indeterminate (DP)	Indeterminate (DP)	Indeterminate (DP)	Indeterminate (DP)	Indeterminate (DP)

b. Permit-overrides: if any rule evaluates to a Permit then it will override any other rule's decision.

	Permit	Deny	NotApplicable	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Permit	Permit	Permit	Permit	Permit	Permit	Permit
Deny	Permit	Deny	Deny	Deny	Deny	Deny
NotApplicable	Permit	Deny	Deny	Deny	Deny	Deny
Indeterminate (D)	Permit	Deny	Deny	Deny	Deny	Deny
Indeterminate (P)	Permit	Deny	Deny	Deny	Deny	Deny
Indeterminate (DP)	Permit	Deny	Deny	Deny	Deny	Deny

c. Deny-unless-permit: sets deny as the default with any permit overriding the deny

	Permit	Deny	NotApplicable	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Permit	Permit	Deny	Permit	Permit	Permit	Permit
Deny	Deny	Deny	Deny	Deny	Deny	Deny
NotApplicable	Permit	Deny	Permit	Permit	Permit	Permit
Indeterminate (D)	Permit	Deny	Permit	Permit	Permit	Permit
Indeterminate (P)	Permit	Deny	Permit	Permit	Permit	Permit
Indeterminate (DP)	Permit	Deny	Permit	Permit	Permit	Permit

d. Permit-unless-deny: sets permit as the default with any deny overriding the permit

	Permit	Deny	NotApplicable	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Permit	Permit	Deny	Permit	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Deny	Permit	Deny	Deny	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
NotApplicable	Permit	Deny	NotApplicable	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Indeterminate (D)	Permit	Deny	Indeterminate (D)	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Indeterminate (P)	Permit	Deny	Indeterminate (P)	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Indeterminate (DP)	Permit	Deny	Indeterminate (DP)	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)

e. First-applicable: accepts the first decision that is either a permit or deny, else indeterminate

	Permit	Deny	NotApplicable	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Permit	Indeterminate	Indeterminate	Permit	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Deny	Indeterminate	Indeterminate	Deny	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
NotApplicable	Permit	Deny	NotApplicable	Indeterminate (D)	Indeterminate (P)	Indeterminate (DP)
Indeterminate (D)	Indeterminate (D)	Indeterminate (D)	Indeterminate (D)	Indeterminate (D)	Indeterminate (DP)	Indeterminate (DP)
Indeterminate (P)	Indeterminate (P)	Indeterminate (P)	Indeterminate (P)	Indeterminate (DP)	Indeterminate (P)	Indeterminate (DP)
Indeterminate (DP)	Indeterminate (DP)	Indeterminate (DP)	Indeterminate (DP)	Indeterminate (DP)	Indeterminate (DP)	Indeterminate (DP)

f. Only-one-applicable: only evaluated if only one valid decision has been made

4 Consent UI / PIP / DB User Guide

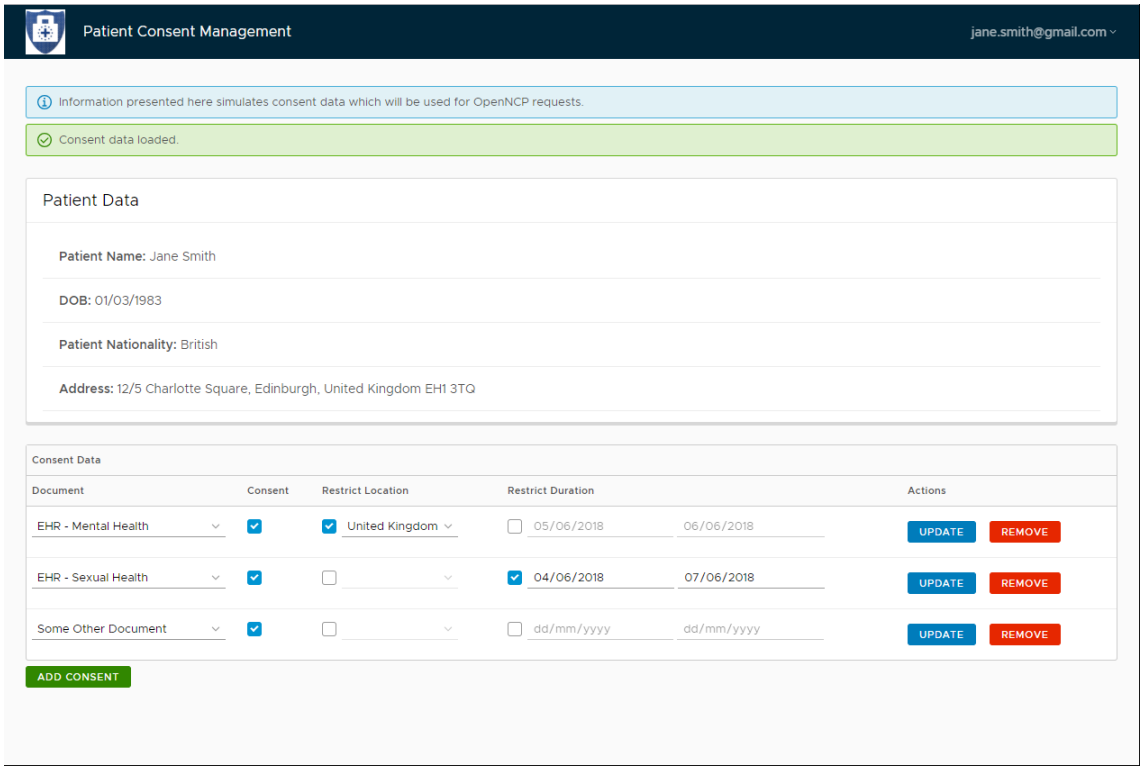

Figure 13: Consent UI / PIP / DB

4.1 UI Fields and Controls

Document: This is the document which the patient wishes to share and can be chosen from a drop-down list of the available documents.

Consent: This indicates that the patient has given consent to share the specified document in some form. If consent is not checked then the document will not be shared.

Restrict Location: If restrict location is not checked then this indicates the patient wishes to share the specified document with all SHEiLD participating nations. If restrict location is checked then the document will only be shared in the specified location.

Restrict Duration: If restrict duration is not checked this indicates that the patient wishes to share the specified document indefinitely.

Action – Update: Clicking the update button saves any changes made to the corresponding consent.

Action – Remove: Clicking remove permanently deletes / removes a consent from the system.

4.2 Usage Examples.

4.2.1 Add a new simple consent

- Click the Add Consent button.
- A new blank consent will be presented on the UI.
- Fill out all of the fields the meaning of each is defined in the previous section.
- Click the update button to save the consent.

4.2.2 Add a consent for multiple locations.

Consent to share a specific document in several different unique locations can be achieved by following the procedure for adding a simple consent as described in section 4.2.1. This procedure is then repeated for each location. To restrict a document to two locations the user would create two consents for the document one for each location.

4.2.3 Add a consent for multiple locations.

Consent to share a specific document in several different time periods can be achieved by following the procedure for adding a simple consent as described in section 4.2.1. This procedure is then repeated for each duration. To restrict a document to two durations the user would create two consents for the document one for each duration.

4.3 Remove a consent

To remove consent for a document without deleting all of the consent data the user can simply un-tick the consent checkbox for the relevant consent.

4.4 Permanently Delete a consent

To permanently delete a consent from the system the user should click the delete consent button for the relevant consent.

5 Summary & Next steps

The components described in the previous sections describe a fully functional consent component. The next phase of the development will focus on the following areas:

1. Refinement of consent management policies defined in the PAP / PDP.
2. Integration with the data sensitivity tool provided by IBM.
3. Integration with the data hiding tool provided by IBM.
4. Integration with OpenNCP.