European Security in Health Data Exchange

Deliverable D5.9

Data hiding tools

<table>
<thead>
<tr>
<th>Editor(s):</th>
<th>Micha Moffie, Muhammad Barham</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsible Partner:</td>
<td>IBM</td>
</tr>
<tr>
<td>Status-Version:</td>
<td>Final - v1.0</td>
</tr>
<tr>
<td>Date:</td>
<td>29/06/2018</td>
</tr>
<tr>
<td>Distribution level (CO, PU):</td>
<td>CO</td>
</tr>
</tbody>
</table>
Project Number: GA 727301
Project Title: SHIELD

Title of Deliverable: Data hiding tools
Due Date of Delivery to the EC: 30/06/2018

Workpackage responsible for the Deliverable: WP5 - Data protection and privacy

Editor(s): IBM
Contributor(s):
Reviewer(s): IBERMATIC
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 Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modifications Introduced</th>
<th>Modified by</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.1</td>
<td>24/04/2018</td>
<td>First draft TOC</td>
<td>IBM</td>
</tr>
<tr>
<td>v0.2</td>
<td>15/05/2018</td>
<td>Draft Package information</td>
<td>IBM</td>
</tr>
<tr>
<td>v0.3</td>
<td>29/05/2018</td>
<td>Draft Introduction</td>
<td>IBM</td>
</tr>
<tr>
<td>V0.4</td>
<td>31/05/2018</td>
<td>Update Introduction, Draft Summary and Next steps</td>
<td>IBM</td>
</tr>
<tr>
<td>V0.5</td>
<td>12/06/2018</td>
<td>Installation instructions</td>
<td>IBM</td>
</tr>
<tr>
<td>V0.6</td>
<td>14/06/2018</td>
<td>User Guide, Executive Summary</td>
<td>IBM</td>
</tr>
<tr>
<td>V0.7</td>
<td>28/06/2018</td>
<td>Comments received from IBERMATICA</td>
<td>IBERMATICA</td>
</tr>
<tr>
<td>V1.0</td>
<td>29/06/2018</td>
<td>Addressing IBERMATICA review comments and finalizing</td>
<td>IBM</td>
</tr>
</tbody>
</table>
# Table of Contents

Table of Contents ................................................................................................................................. 4  
List of Figures .......................................................................................................................................... 5  
Terms and abbreviations .......................................................................................................................... 5  
Executive Summary .................................................................................................................................. 6  
1 Introduction .............................................................................................................................................. 7  
1.1 About this deliverable .......................................................................................................................... 7  
1.2 Progress made ...................................................................................................................................... 7  
1.3 Document structure ............................................................................................................................. 7  
2 Delivery and usage .................................................................................................................................... 8  
2.1 Package information ............................................................................................................................. 8  
2.2 Installation instructions ......................................................................................................................... 9  
2.2.1 WAR ............................................................................................................................................... 9  
2.2.2 Docker .......................................................................................................................................... 10  
2.2.3 Setup ............................................................................................................................................ 10  
2.3 User Guide .......................................................................................................................................... 12  
2.3.1 Rest API ....................................................................................................................................... 12  
2.3.2 Policy example ............................................................................................................................... 13  
2.3.3 UI - support for managing and testing policies ............................................................................. 18  
3 Summary & Next steps ............................................................................................................................ 19  
4 References ............................................................................................................................................... 20
List of Figures

FIGURE 1 - DATA HIDING TOOL - DEPLOYED USING TOMCAT WEB APPLICATION MANAGER.......................... 9
FIGURE 2 - DATA HIDING TOOL - BUILDING A DOCKER CONTAINER ......................................................... 10
FIGURE 3 - DATA HIDING TOOL - SWAGGER UI .................................................................................. 13
FIGURE 4 - DATA HIDING TOOL - PAYLOAD EXAMPLE FOR SHIELD ...................................................... 14
FIGURE 5 - DATA HIDING TOOL - POLICY EXAMPLE FOR SHIELD ....................................................... 14
FIGURE 6 - PAYLOAD INPUT .................................................................................................................. 17
FIGURE 7 - PAYLOAD OUTPUT ......................................................................................................... 17
FIGURE 8 - DATA HIDING TOOL - POLICY AUTHORIZING UI - UNDER CONSTRUCTION ..................... 18

Terms and abbreviations

<table>
<thead>
<tr>
<th>EC</th>
<th>European Commission</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDPR</td>
<td>General Data Protection Regulation</td>
</tr>
</tbody>
</table>
Executive Summary

This deliverable is the second deliverable related to Task 5.4: Data hiding tools and aims at providing support for initial evaluation of the architecture and functionality.

This deliverable has two parts: a prototype and a document containing installation instruction and a user guide. In addition, this document reports on the main progress made in the prototype: Providing support for policy persistency, improve and extend the REST API's, introducing a new Graphical User Interface, supporting advanced deployment methods and applying the tool to Shield use case example.
1 Introduction

1.1 About this deliverable

This deliverable is the second deliverable in Task 5.4: Data hiding tools. The main part of the deliverable is the prototype. In addition, the deliverable includes this document which report on the progress that has been made in the development of the tool and provide initial documentation for the adaption and integration of data hiding tool within Shield.

1.2 Progress made

We build upon the work done in previous deliverables including Deliverable D5.4 – data hiding tools [1], Deliverable D6.1 - use case specification [2] and Deliverable D2.3 - SHiELD Architecture [3] and made progress in the following areas:

1. Provide support for policy persistency: extend the tool’s service to store policies. The persistency is realized using Cloudant - a highly scalable and performant NoSQL JSON Document store [4] which is compatible with Apache CouchDB [5].

2. Tool’s service REST API improvements and extensions: (a) encapsulating the policy within a JSON object to allow for flexibility and extendibility such as adding a policy name. (b) Adding an API to search policies (e.g. based on name). And (c) simplifying the API by removing the concept of Context (see [6]).

3. Graphical User Interface for managing and testing policies. Developed a draft GUI able to support the management of policies as well as testing and execution of policies on different payloads.

4. Apply the data hiding tool to Shield by preparing a policy able to mask a clinical document example, compatible to the epSOS format provided by the use cases.

5. Support advanced deployment methods including (a) the creation of a WAR file to be dropped in any application server (e.g. Apache Tomcat) (b) creation of a standard docker container to support a process of deploying the container within a container service such as kubernetes [7].

1.3 Document structure

The main content is presented in the next section. There we provide documentation about the packaging, delivery and usage of the tool. We cover information related to deployment as well as usage instructions. (Note, for detail about the tool implementation details see Deliverable D5.4 [1]). The last section provides a short summary and next steps.

---

1 Cloudant was selected to support IBM exploitation efforts. The tool can support any Compatible NoSQL store identified by WP2.
2 Delivery and usage

2.1 Package information

For convenience and flexibility, we provide two ways to deploy the data hiding tool:

1. A WAR file – the standard Web application Archive to be dropped in an application server (e.g. Apache Tomcat).
2. A Docker container - a standard (Linux) container to be instantiated in container services such as Cloud Foundry, Microsoft Azure, Amazon AWS etc.

The WAR file contains all the necessary code and dependent libraries to realize the exposed API’s. The WAR file follows the standard structure containing the META-INF and WEB-INF directories as well as the deployment descriptor (web.xml).

In addition, the deployment contains Swagger UI, which is configured to present and support interactions with the data hiding tool REST API’s. Moreover, the deployment will include an initial UI which can be used to manage and test policies.

As explained above, we also support deploying a Docker container. The deployed container includes a web application server already configured to execute the service. (effectively – a web application server with the WAR file deployed). We support creating a docker container running the tomcat apache server.
2.2 Installation instructions

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

2.2.1 WAR

To install the WAR file, one simply needs to deploy the WAR file in the web server. For example, in Tomcat it can be copied to the webapps/ directory or uploaded using the Tomcat Web Application Manager (see Figure 1).

![Tomcat Web Application Manager](image)

**Figure 1 - Data Hiding Tool - Deployed using Tomcat Web Application Manager**

Once deployed, the application should be started.
2.2.2 Docker

We have build a docker image out of the WAR file. Once the image is build it is stored in docker’s repository.

Building a docker container is done using a docker file. An example of a docker file shown here:

```
FROM tomcat:8.0.20-jre8
RUN mkdir -p /usr/local/tomcat/conf
COPY ./magen.war /usr/local/tomcat/webapps/magen.war
COPY ./tomcat-users.xml /usr/local/tomcat/conf/
```

*Figure 2* shows the command and the output of building a docker file.

![Image of building a docker container](image)

*Figure 2 - Data Hiding Tool - Building a Docker container*

Running the container can be done, for example using the command `docker run -p 80:50002 magen-service`. This command will run the docker image (and map port 50002 in the container to the host port 80).

2.2.3 Setup

Currently, the user can configure the following aspects of the application:

1. The port number where the service is provided by the web server
2. The information necessary to persist policies on cloudant, namely: url, database, username and password.

The application allows for configuration to be provided through a file (application.properties) as well as environment variables (usually used for docker containers).
An example of an application properties file is presented here:

```properties
# server port
server.port=50002

# cloudant configuration
cloudant.url=https://mycloud.cloudant.com
cloudant.username=testusername
cloudant.password=2aeda830b9e189008215c76027977629e8f8d5b4
cloudant.db=magen
```

Environment variables can be set in the following way:

```bash
CLOUDANT_URL https://mycloud.cloudant.com
CLOUDANT_USERNAME testusername
CLOUDANT_PASSWORD 2aeda830b9e189008215c76027977629e8f8d5b4
CLOUDANT_DB magen
```
2.3 User Guide

The data hiding tool provides a rest service with two main types of API.

1) Policy management – CRUD (Create, Read, Update and Delete) API’s for policy objects, and
2) Process – A single API able to process a payload.

The policy management operations are performed infrequently – one can consider these to be done during design time – when the data flow is being setup or modified. The process API will be called frequently – during run time – every time a payload is transferred from one location to another. When this happens, the user is expected to call the service with the payload itself, the policy id, the keys if encryption or decryption are necessary as well as additional metadata such as predicated (when required by the policy).

2.3.1 Rest API

A full description of the Rest API’s parameters is provided as part of the data hiding tool in Swagger. Figure 3 shows the hiding tools Swagger API².

² ‘Magen’ is our internal project name for the data hiding tool. In Hebrew, Magen means shield.
In this section we present an example policy created to address the use case requirement. For an overview on policies we refer the reader to Deliverable D5.4 – data hiding tools [1].

In the first stage, the use case requirements were to mask (encrypt or redact) an XML payload – the epSOS patient summary xml. Figure 4 shows the XML elements that are required to be masked. Note that these include both XML Elements as well as XML attribute values.
We have constructed a policy able to encrypt and decrypt the required elements and attributes in the payload. The policy itself is setup with an XML processor and XPath Selectors for each element or attribute. It contains an ‘any processor’ which is used to process a predicate allowing the user to specify during run time whether to encrypt or decrypt the payload. Lastly it contains the encryption and decryption processors to perform the encryption and decryption operation. Figure 5 shows the policy created.

![Figure 4 - Data Hiding Tool - Payload Example for Shield](image)

**Figure 5 - Data Hiding Tool - Policy Example for Shield**
Following is the complete policy in json

```json
{
    "name": "Osakidetza.v1",
    "body": {
        "version": "1.0",
        "graph": {
            "nodes": [
                {
                    "id": "xmlProcessor",
                    "root": true
                },
                {
                    "id": "anyProcessor"
                },
                {
                    "id": "encryptionProcessor"
                },
                {
                    "id": "decryptionProcessor"
                }
            ],
            "edges": [
                {
                    "source": "xmlProcessor",
                    "target": "anyProcessor",
                    "selector": "XPathIdSelector"
                },
                {
                    "source": "xmlProcessor",
                    "target": "anyProcessor",
                    "selector": "XPathAddressSelector"
                },
                {
                    "source": "xmlProcessor",
                    "target": "anyProcessor",
                    "selector": "XPathPhoneSelector"
                },
                {
                    "source": "anyProcessor",
                    "target": "encryptionProcessor",
                    "selector": "anySelector",
                    "predicate": "encryptionPredicate"
                },
                {
                    "source": "anyProcessor",
                    "target": "decryptionProcessor",
                    "selector": "anySelector",
                    "predicate": "decryptionPredicate"
                }
            ]
        },
        "processors": [
            {
                "id": "xmlProcessor",
                "type": "XmlProcessor",
                "selector": "XPathIdSelector",
                "predicate": "encryptionPredicate"
            }
        ]
    }
}
```
"loggable" : true
},
{
"id": "anyProcessor",
"type": "AnyProcessor",
"loggable" : true
},
{
"id": "encryptionProcessor",
"type": "EncryptionProcessor",
"configuration" : { "keysize" : 16, "mode": "encrypt" },
"loggable" : true
},
{
"id": "decryptionProcessor",
"type": "EncryptionProcessor",
"configuration" : { "keysize" : 16, "mode": "decrypt" },
"loggable" : true
}
],
"selectors": [  
{
"id": "XPathIdSelector",
"type" : "XpathSelector",
"configuration" : { "xpath" : "*[name()='ClinicalDocument']/*[name()='recordTarget']/*[name()='patientRole']/*[name()='id']/@root | *[name()='ClinicalDocument']/*[name()='recordTarget']/*[name()='patientRole']/*[name()='id']/@extension | *[name()='ClinicalDocument']/*[name()='recordTarget']/*[name()='patientRole']/*[name()='id']/@nullFlavor" },
"loggable" : true
},
{
"id": "XPathAddressSelector",
"type" : "XpathSelector",
"configuration" : { "xpath" : "*[name()='ClinicalDocument']/*[name()='recordTarget']/*[name()='patientRole']/*[name()='addr']/node()" },
"loggable" : true
},
{
"id": "XPathPhoneSelector",
"type" : "XpathSelector",
"configuration" : { "xpath" : "*[name()='ClinicalDocument']/*[name()='recordTarget']/*[name()='patientRole']/*[name()='telecom']/*[name()='value']/@value" },
"loggable" : true
},
{
"id": "anySelector",
"type": "AnySelector",
"loggable": true
}]}
"predicates": [
  {
    "id": "encryptionPredicate",
    "type": "ExistPredicate",
    "configuration": {"key": "encrypt"},
    "loggable": true
  },
  {
    "id": "decryptionPredicate",
    "type": "ExistPredicate",
    "configuration": {"key": "decrypt"},
    "loggable": true
  }
]

Next, we show the relevant part of the payload where masking was applied. Figure 6 shows the payload input while Figure 7 shows the output (encrypted).

```
<patientRole>
  <id root="2.16.724.4.16.1.100.2.1" extension="10574682" />
  <id root="2.16.724.4.40" extension="88888888888888888888" />
  <nullFlavor="UNF" />
  <id root="2.16.724.4.41" extension="882494" />
  <nullFlavor="UNF" />
</patientRole>
```

**Figure 6 - Payload Input**

```
<patientRole>
  <id extension=""#NUYo9qWzpjDBjkRSmO0Q=="" root=""#ac9Bhy#aa5+9V61ZV8rRyp0CEK9obY2lh8SpU=#"" />
  <id extension=""#9B1R/vmU19Ou08S5kA92LzVWPmG8h6odXxwht7=t?#/" root=""#4owKh1RyjIjy8p9Q8iV3w=="" />
  <id extension=""#9B1R/vmU19Ou08S5kA92LzVWPmG8h6odXxwht7=t?#/" root=""#4owKh1RyjIjy8p9Q8iV3w=="" />
  <nullFlavor=""#hYJbRydVdp9iI97oKjFVwD9687VjemyhjWx9lx79n=="" />
  <nullFlavor=""#hYJbRydVdp9iI97oKjFVwD9687VjemyhjWx9lx79n=="" />
</patientRole>
```

**Figure 7 - Payload Output**

Project Title: SHIELD

Contract No. GA 727301

www.project-shield.eu
2.3.3 UI - support for managing and testing policies

As mentioned above, the deployment includes an initial UI which can be used to manage and test (run) policies. It is currently in draft and we will be modified & improved soon.

Figure 8 shows the hiding tools draft UI. Please note that it is currently under construction and can significantly change.

Figure 8 - Data Hiding Tool - Policy Authoring UI - under construction
3 Summary & Next steps

We have made advancements in several fronts. First, in terms of functionality, we made improvements and extensions to the service Rest API's, provided persistency of policies and introduced a GUI (draft version) for managing and testing policies. Second, in terms of adaption to Shield, we created a policy and successfully applied the tool to the use case (Clinical Document in epSOS format). Lastly, we made advancements in the area of deployment of the service to support both simple (WAR) and advanced deployment methods based on docker containers which can be used to deploy on Kubernetes.

We plan to continue to advance in these fronts. Specifically, we plan to advance the functional capabilities of the service by improving the GUI to support the creation of policies and extend our masking library by adding specialized processors and selectors able to address use case requirements for unsupported formats. We will continue to work with the use case providers to make sure the tool is successfully applied, addresses the use case requirements and provides support for adhering to the GDPR regulations. Lastly, we plan to continue the work with SHiELD architects to integrate and deploy the data hiding tool within the OpenNCP Enforcement Point.
4 References