# An E-government Interoperability Platform Supporting Personal Data Protection Regulations

**Laura González, Andrés Echevarría, Dahiana Morales, Raúl Ruggia**
Instituto de Computación, Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay, 11300
*{lauragon, juan.echevarria, dahiana.morales, ruggia}@fing.edu.uy*

**Abstract**

Public agencies are increasingly required to collaborate with each other in order to provide high-quality e-government services. This collaboration is usually based on the service-oriented approach and supported by interoperability platforms. Such platforms are specialized middleware-based infrastructures enabling the provision, discovery and invocation of interoperable software services. In turn, given that personal data handled by governments are often very sensitive, most governments have developed some sort of legislation focusing on data protection. This paper proposes solutions for monitoring and enforcing data protection laws within an E-government Interoperability Platform. In particular, the proposal addresses requirements posed by the Uruguayan Data Protection Law and the Uruguayan E-government Platform, although it can also be applied in similar scenarios. The solutions are based on well-known integration mechanisms (e.g. Enterprise Service Bus) as well as recognized security standards (e.g. eXtensible Access Control Markup Language) and were completely prototyped leveraging the SwitchYard ESB product.

**Keywords:** data protection, privacy, e-government, enterprise service bus, eXtensible Access Control Markup Language, interoperability.

## 1 Introduction

During the last decades, many governments have driven e-government initiatives with the goal of improving the quality of public services offered to citizens [1]. To this end, governments have implemented e-government systems which enable a more efficient inter-organizational coordination between public agencies. These systems often rely on interoperability platforms which provide a hardware and software infrastructure in order to facilitate the interconnection between the software systems operating in these agencies [2].

In Uruguay, for example, the Electronic Government and Information Society Agency (AGESIC[1], Agencia de Gobierno Electrónico y Sociedad de la Información) have made available an Interoperability Platform (InP), as part of the Uruguayan E-Government Platform (EGP) [3]. The InP provides infrastructure and utility services in order to decrease the complexity of developing e-government services for the citizens or for other public agencies. This platform is also the foundation for implementing a state-wide Service Oriented Architecture (SOA) [4] in which services offered by public agencies are described, published, discovered, invoked and combined using standard protocols and interfaces.

On the other hand, given that governments handle citizens' personal data, which may be highly sensitive, many countries have developed some sort of legislation focused in personal data protection [5]. In particular, the Personal Data Protection and "Habeas Data" Action Act (Act 18.331 [6]) of Uruguay establishes that the right to the protection of personal data is inherent to the person. The act specifies a set of personal data which are public (e.g. names, last names, national identification number) and establishes that the rest are private or sensitive. Moreover, if a public agency wants to use sensitive personal data or share them with other agencies, it needs to obtain the explicit consent of the involved citizens.

Since developing mechanisms to ensure the compliance with personal data protection laws may be complex and costly for public agencies, it would be convenient that e-government interoperability platforms provide mechanisms which allow managing, monitoring and enforcing this type of regulations without involving ad-hoc programming in business (i.e. e-government) applications. While the ideal approach would consist in performing the enforcement

[1] http://www.agesic.gub.uy/

using only platform's mechanisms, this approach poses a number of challenges related to monitoring inter-agency message interactions, detecting potential regulation violations and dynamically transforming messages to be compliant with regulations. This may explain the lack of implementations based on this approach.

This paper addresses these issues and proposes an extended e-government interoperability platform to monitor and enforce data protection regulations in inter-agency interactions through platform's mechanisms. Among others, these mechanisms include dynamic adaptability capabilities developed in previous work [7][8][9]. The application scenario of this work is the Uruguayan InP and Data Protection regulations [6] although it may be applied in other platforms and laws with similar characteristics. The proposed extensions are based on widely established integration mechanisms, like the Enterprise Service Bus (ESB) [10], and on recognized security standards, like the eXtensible Access Control Markup Language (XACML) [11]. The solution was completely prototyped using the SwitchYard[2] ESB product and was evaluated through the development of case studies and response time tests. A previous version of this work was presented in [12].

The rest of the paper is organized as follows. Section 2 presents background on topics associated with the proposal. Section 3 analyses and identifies the requirements that were addressed. Section 4 describes the proposed solution to deal with the identified requirements. Section 5 presents implementation and experimentation details. Section 6 analyses related work and, finally, Section 7 presents conclusions and future work. Also, a list of abbreviations is included in Appendix A.

## 2 Background

This section briefly describes technologies, standards and topics which are relevant for the proposal.

### 2.1 Web Services

Web Services are software applications identified by a URI. Their interfaces and access methods are defined, described and discovered as XML artefacts. Web Services enable to implement direct interaction between software components by using XML messages which are exchanged through Internet-based protocols [13].

Web Services have become the mainstream technology for implementing Service Oriented Architectures (SOA) and they are the main mechanism to integrate multi-platform software applications [13].

The main standards supporting the Web Services technology are: Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL). Other standards (e.g. WS-Security, WS-Addressing) extend the former ones to support advanced features.

SOAP [14] uses an XML-based format to build messages independently from the underlying transport protocol and provides mechanisms for specifying how messages have to be processed. SOAP messages consist of an envelope with a *header* and a *body*. The message header is extensible and may contain several elements to specify different type of information, for example related to security and addressing. The most popular transport protocol for SOAP messages is HTTP, but the standard is not restricted to it.

WSDL [15] is an XML-based language to specify service interfaces, which enables to describe Web Services in an standard way. WSDL documents consist of two main parts: an abstract description and a concrete description. The abstract description includes elements to specify functional aspects and those involving to data structures related to Web Service operations. The concrete description includes elements that: (i) provide instructions to interact with the Web Service through a specific protocol (e.g. SOAP over HTTP), and (ii) specify a concrete network address to invoke it.

Web Services Addressing (WS-Addressing) [16] focuses on features related to the message processing and delivery. To this, it defines elements that enable to specify these characteristics independently from the transport protocol. For example, the element "wsa:to" enables to specify the message destination, while the element "wsa:action" allows to specify its semantics.

Web Services Security (WS-Security) [17] specifies extensions to SOAP enabling to ensure the integrity, confidentially and authentication of the messages. In particular, WS-Security describes how to include and use security tokens in SOAP messages. The *UserNameToken*, for instance, enables to specify a username and optionally a password. On the other hand, XMLTokens enable to attach XML-based security tokens using different formats such as the Security Assertion Markup Language (SAML).

### 2.2 Enterprise Service Bus

An Enterprise Service Bus (ESB) is a standards-based integration platform which combines Web Services, data transformation and intelligent routing in order to reliable implement the interaction between software components with transactional integrity [10].

[2] http://switchyard.jboss.org/

ESBs provide an intermediate layer with reusable integration capabilities in order to enable the interaction between clients and services in a SOA. ESBs receive message-based requests on which they perform mediation operations to overcome client-server heterogeneities [10].

Using ESBs promotes a loosely coupling between clients and services by providing the means to use large-scale business logic through modular services which may be independently invoked. This also enables to separate the integration logic, the communication logic and the business logic implemented by services [10]. This way, different applications and services implemented on different technology stacks and using different data formats and protocols may communicate with an ESB through well-defined service interfaces. In turn, these services may be orchestrated and used by other applications and services.

The most relevant ESB functionalities for the purpose of this work are message transformation, enriching and routing [10]. ESB products provide mechanisms to transform and enrich messages exchanged between clients and services by using, for instance, the XSLT standard [18]. These transformation mechanisms may be used to solve different problems such as heterogeneity of data formats. In addition, ESBs have capabilities to define at runtime a message destination using different criteria. The main routing types are content-based and itinerary-based routing. Content-based routing determines the message destination based on its content, for example, using the content in the header or body of SOAP messages [9]. On the other hand, the itinerary-based routing (also known as Routing Slip) determines message destination taking as input an itinerary description, which may be included in the message [10].

### 2.3 eXtensible Access Control Markup Language

XACML [11] is an OASIS standard specification which describes a language for defining access control policies as well as a language to request and response access control decisions in XML. Usually, a requester tries to run an action on a resource by sending a request to the component that protects it: Policy Enforcement Point (PEP). The PEP performs an authorization request based on attributes of the requester, the resource, the action to be executed and any other relevant information. This request is sent to a Policy Decision Point (PDP) which issues a response indicating if access should be allowed based on the request and policies managed through a Policy Administration Point (PAP). Reponses consist of one of the following values: Permit, Deny, Indeterminate (some error occurred) and Not Applicable (the request cannot be responded by this service). Also, a Policy Information Point (PIP) may be used if additional information for taking the authorization decision is needed. Based on the authorization response, the PEP allows or denies access to the requester.

Figure 1 presents an XACML information flow which shows the different participating components: Policy Administration Point, Policy Decision Point, Policy Enforcement Point, Policy Information Point and Context handler.

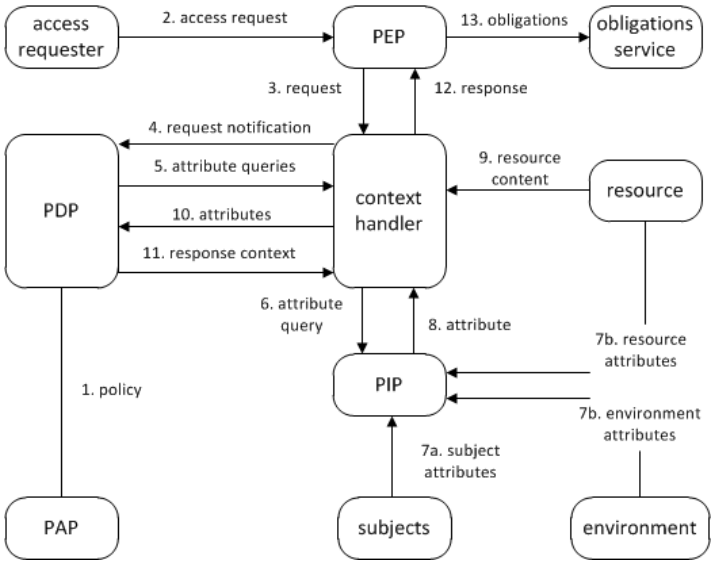


**Figure 1:** XACML information flow [11]

The Policy Administration Point (PAP) is the component which creates policies and sets of policies. The Policy Decision Point (PDP) is the component which evaluates the policies and issues an authorization decision. The Policy Enforcement Point (PEP) is the component which executes the access control performing authorization requests and enforcing the responses. The Policy Information Point (PIP) is the component which works as a source of attributes.

Finally, the Context Handler transforms the requests from the native format to the canonic XACML format and the authorization decisions from the canonic XACML to the native format.

Figure 2 presents an example of XACML request where Juan Perez (*subject*) requests an authorisation to read (*action*) his medical records (*resource*).

```
<Request>
    <subject>Juan Perez</subject>
    <resource>file://example/med/record/patient/Juan_Perez</resource>
    <action>Read</action>
    <environment></environment>
</Request>
```

**Figure 2:** Example of XACML Request

In turn, Figure 3 presents the response to the former request in which the decision is to permit (*decision*) the access to the records (*resource*) as requested by the subject (*subject*) for read (*action*).

```
<Response>
    <decision>Permit</decision>
    <status>
        <statusCode>ok</statusCode>
        <statusMessage></statusMessage>
    </status>
    <obligations></obligations>
</Response>
```

**Figure 3:** Example of XACML Response

### 2.4 E-government Platforms

It has been more than a decade since e-government systems were recognized as strategic enablers for improving the efficiency and quality of public services delivered to citizens. Nowadays, e-government systems mainly consist of countrywide infrastructure providing self-services and relevant information to citizens, implementing shared governmental services such as e-Identity and enabling a more effective and reliable inter-organizational coordination among public agencies and partners [1]. More recently, the United Nations has linked e-government to achieving sustainable development and the Millennium Development Goals (MDGs) as *"E-government and innovation can provide significant opportunities to transform public administration into an instrument of sustainable development."* [19].

In this context, e-government platforms have become a key tool to support the development of e-government in many countries. Usually based on middleware technologies, such platforms provide the means to interconnect information systems of public agencies, provide common services that generate economy of scale, and foster the implementation of multi-agency services [2][3].

In order to facilitate the integration between agencies that may have different technological environments, e-government platforms are based on standards. Beyond their particular characteristics, these platforms usually provide a set of basic capabilities which include, among others, security (e.g. authentication), interoperability (e.g. through the use of standards) and mediation services implementing Enterprise Integration Patterns (EIP), e.g. data transformation [20]. More concretely, their mediation and interoperability capabilities are usually provided by middleware technologies, such as SOAP Web Services and Enterprise Service Bus (ESB). In addition, security capabilities usually rely on well-established standards such as XACML [11].

While the traditional focus of e-government (systems and platforms) have been the implementation of G2C (government-to-consumer), G2G (government-to-government) and G2B (government-to-business) interactions, new trends for including private organizations providing public services (e.g. health services providers) are leading to expand e-government models. As a consequence, new generation e-government platforms have to include functionalities to deal with a wider set of organizations participating in the platform services.

### 2.5 Data Protection Laws

Data Protection regulations establish legal rules on how personal data should be managed and used by public and private organizations in order to protect individuals' privacy and to ensure an appropriate quality for these data.

The increasing implementation of online services, notably in e-government, in which individual's personal data is used to carry out a transaction has made necessary to consider Data Protection enforcement as a key component of e-government systems and platforms. Furthermore, as government is maintaining ever larger stores of personal

information, the risk of privacy invasion by governments increases. Personal data handled by governments may be very sensitive [5] and their analysis can be highly invasive when data is combined and aggregated.

As a result, most governments have promulgated Data Protection laws [1][21]with different approaches [5], which mainly rule on the re-use of information in other contexts from which it was provided.

In turn, the OECD has developed Guidelines on the Protection of Privacy and Transborder Flows of Personal Data aiming at "*harmonising national privacy legislation and, while upholding such human rights, would at the same time prevent interruptions in international flows of data*" [22]. OCDE Guidelines establish a set of basic principles, which correspond to the ones in many country regulations: collection limitation, data quality, purpose specification, use limitation, security safeguards, openness, individual participation and accountability.

In many countries, one of the key mechanisms for Data Protection are the *explicit consents* that citizens have to provide in order to allow the agencies to use / share their personal data for a given purpose and within a given time period [6][23]. This way, agencies are required to comply with these consents when using and sharing information between them. In addition, many countries also provide habeas data actions which give citizens the right to access and correct their personal data and the right to a judicial hearing in the matter of personal data protection [24]. These actions establish specific deadlines for the agencies once a citizen has requested access or a correction to their data.

Although Data Protection regulations have been established since many years, their implementation in e-government scale is still very limited.

## 3 Requirements Analysis

This section analyses the general context of this work and identifies the requirements to be met by an interoperability platform in order to manage, monitor and enforce data protection regulations. As stated before, this work focuses on the Uruguayan InP (described in Section 3.1) and on the Uruguayan data protection regulations (described in Section 3.2). However, as explained in Section 3.4, the identified requirements (presented in Section 3.3) may also apply to other interoperability platforms and data protection regulations with similar characteristics.

### 3.1 Uruguayan Interoperability Platform

The Uruguayan Interoperability Platform (InP) has the goal of facilitating and promoting the development of e-government services in Uruguay [3]. The platform has two main components: the Middleware Infrastructure and the Security System.

The Middleware Infrastructure (MI) provides mechanisms that facilitate the development, deployment and integration of services and applications. These mechanisms are also the foundation for implementing an state-wide Service Oriented Architecture (SOA). Indeed, public agencies can leverage this infrastructure to publish and consume services. In addition, they can use its mediation capabilities, which are mainly provided by an ESB, in order to decouple clients and services.

The Security System (SS) provides security services to the rest of the components in the InP and it is the responsible for enforcing the required authentication, authorization and auditing policies. In particular, the SS provides mechanisms that allow controlling the access to the services published in the InP.

The services offered by public agencies through the InP are exposed using the Web Services technology. Each Web Service has a set of operations which receive and return a set of elements as input and output parameters, respectively. Figure 4 presents a conceptual model with these notions.

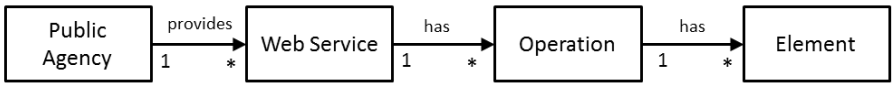


**Figure 4:** Conceptual Model of the Web Services exposed through the InP

For example, the Ministry of Industry, Energy and Mining (Ministerio de Industria, Energía y Minería, MIEM) provides a Web Service named "CertificadosService" [3] which offers two operations: "getCertificadoPymeByRUT" and "getCertificadoCCPByCodigo". Some elements of the operation "getCertificadoPymeByRUT" are: "rut", "code", "desc", "razonSocial", "tipoDeSociedad" and "vigencia".

When a public agency wants to invoke a Web Service operation through the InP, it should send a SOAP message to this platform with the data required by the service (e.g. input parameters, security information). Once this message arrives at the platform, it is routed to the SS which performs access control tasks with the aim of allowing or denying the invocation of the operation. In order to take this authorization decision, the SS needs that

[3] http://www.agesic.gub.uy/innovaportal/v/3392/1/agesic/dinapyme_consulta_de_certificados.html

the invoking agency includes in the SOAP message the identification of the service and the name of the operation it wants to invoke. These items have to be included using the WS-Addressing standard through the following elements: “wsa:to”, to specify the service, and “wsa:action”, to specify the operation. In addition, the invoking agency must include a security token in the message which has to be previously obtained from the InP, using the WS-Trust standard. This token, which among other elements contains the invoking agency and the role of the invoking user, has to be included in the message as an XML Security Token using the WS-Security standard.

After these security controls are completed, the message is routed to the MI where some validations (e.g. data format validations) and, if required, transformations (e.g. including a missing element in the message) are performed. Finally, the message is sent to the target service which is hosted in the servers of a public agency.

Figure 5 depicts the process of a Web Service invocation within the InP. In particular, it shows how the National Direction of Civil Identification (Dirección Nacional de Identificación Civil, DNIC) invokes a Web Service provided by the Social Security Institute (Banco de Previsión Social, BPS).

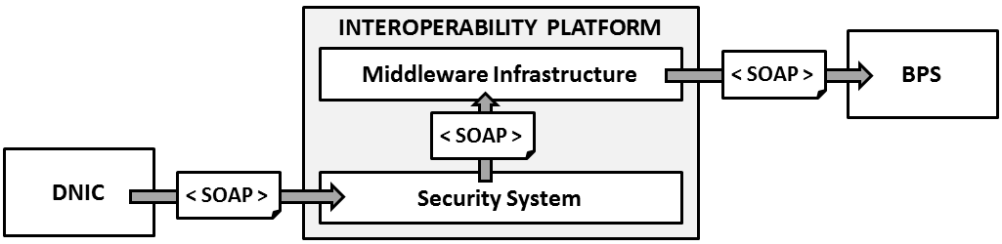


**Figure 5:** Service Invocation in the InP

### 3.2 Uruguayan Data Protection Law

The Personal Data Protection and “Habeas Data” Action Act [6] of Uruguay establishes that the right to the protection of personal data is inherent to the person (natural or legal). Personal data may include sounds, images or biometric data, among others. Some examples of personal data are names, last names, e-mails, pictures, fingerprints, voice and ADN. The act specifies a set of personal data which are public, that is, it is not required to obtain the explicit consent of the data owner to manipulate them. For natural persons these data are: names, last names, national identification document, nationality, address and birthdate [6].

The act also define “sensitive data” as personal data which reveals the racial origin, ethnic origin, political preferences, religious or moral convictions, union affiliation or information related to the health or sexual life of a person. The act establishes that no one is obligated to provide such data and if an entity wants to obtain them it must have the explicit consent of the data owner. This consent must be free, that is, the person should provide it in a voluntary way and for a given time period.

On the other hand, the act establishes that data owners have the right to know which data about them each entity (e.g. a public agency) has. To this end, data owners can request this information to the different entities. This right can be executed every six months and the requested information has to be provided within a period of five business days. The information can be provided in writing or by electronic means.

Through the analysis of this regulation, the following concepts were identified:

- Personal Datum: Information of any type concerning natural or legal persons which are identified or identifiable.
- Citizen: In the context of this work, Citizen refers to a natural person. Even though the act also deals with legal persons, they were left out of the scope of this work.
- Purpose: It refers to the finality for which personal data will be used. For instance, personal data can be used by a public agency for a specific e-government procedure.
- Consent: It refers to the permission given by a citizen to an entity (e.g. public agency) to share its sensitive personal data with another agency. Although the act also refers to consents for using personal data, this work focuses on consents for sharing personal data given that these are the ones which can be monitored through an interoperability platform. Consents are given for a specific purpose and for a given time period.

Figure 6 presents these concepts as well as their relationships. Briefly, Citizens provide Consents to Public Agencies to share their Personal Data with other Public Agencies for a given Purpose.

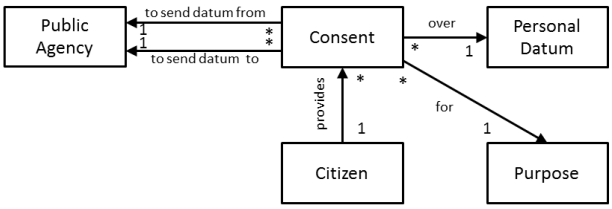


**Figure 6:** Conceptual Model of the Data Protection Regulation

### 3.3 High Level Requirements

The analysis of the general context of this work, presented in Section 3.1 and Section 3.2, allows identifying various high level requirements for the solution, which should extend the InP to support data protection regulations.

The solution has to monitor the messages (for invoking services) exchanged through the InP between public agencies and enforce the compliance with the personal data protection law. To this end, the solution has to intercept all the exchanged messages and perform the required validations in order to determine if messages met what the law establishes. The validation have to be performed considering the public agencies that participate in the message exchange, the citizen who owns the data that is being exchanged, the purpose of the data exchange, the date in which messages are sent and the consents provided by the citizen.

In addition, the solution has to provide tools that allow citizens to manage the consents they provide to the different public agencies. Through these tools, citizens have to be able to obtain the personal data that each public agency has about them, under the terms and forms prescribed by the law. The solution also has to allow public agencies to manage the different personal data requests performed by the citizens.

Lastly, the solution should provide a configuration tool which allows administrators to manage and monitor all the aspects of the solution. For example, this tool should allow configuring the participating public agencies, the exposed services, the mapping between the elements of service operations and personal data, the actions to be taken in case a message exchange is not compliant with the law, among others.

Figure 7 presents a consolidated conceptual model for the solution where all the concepts identified in Section 3.1 and Section 3.2 are included. In addition, according to the analysis of this section the model was enhanced with:

- Action: It is a task to be performed in case of detecting that some of the aspects of the personal data protection regulation are not being met.
- "has associated": A relationship between Operation and Action which specifies the actions to be taken in case the message exchanges to invoke the operation are not compliant with the law.
- "uses": A relationship between Purpose and Operation which indicates which operations are required for a given purpose (e.g. an e-government procedure)
- "associated with": A relationship between Element and Personal Datum which maps the elements of operations (i.e. input and output parameters) with personal data.

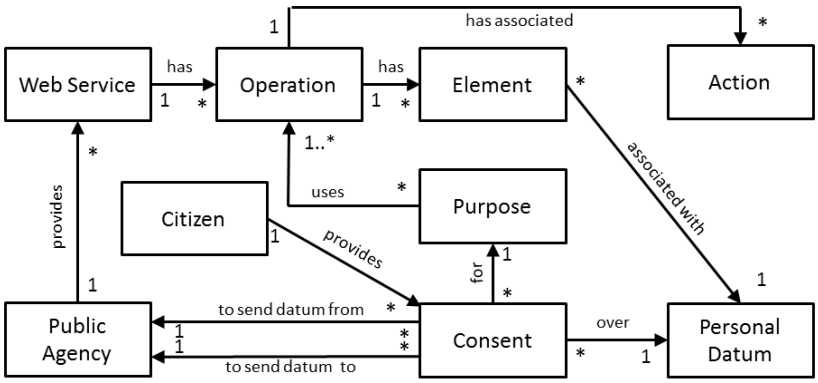


**Figure 7:** Consolidated Conceptual Model

### 3.4 Detailed Requirements

From the high level requirements described in Section 3.3, three types of users were identified to which the solution should provide functionalities:

- General Administrators
- Public Agency Administrators
- Citizens

General Administrators have to be able to manage all the aspects of the solution. Table 1 lists and describes the specific requirements for this type of users.

**Table 1:** Requirements for General Administrators

| ID | Name | Description |
|---|---|---|
| RQGA-1 | Public Agencies Management | The solution has to allow managing (i.e. adding, modifying, removing and listing) the public agencies that are integrated in the interoperability platform. |
| RQGA-2 | Users and Roles Management | The solution has to allow managing the users and roles (e.g. citizen, public agency administrator) which use the solution as well as configuring the access they have to the different functionalities of the platform. |
| RQGA-3 | Personal Data Management | The solution has to allow managing the different personal data (e.g. name) handled by the platform as well as specifying if the data are public or sensitive. |
| RQGA-4 | Purposes Management | The solution has to allow managing the purposes handled by the platform as well as associating which operations are required for each one. |
| RQGA-5 | Actions Management | The solution has to allow managing the actions to be performed in case personal data are exchanged without having the required consents. |
| RQGA-6 | Services Management | The solution has to allow managing the services and operations which are provided by public agencies and are exposed through the platform. It also has to allow associating the actions to be taken when an invocation to an operation is not compliant with the law. |
| RQGA-7 | Consents Management | The solution has to allow managing the consents provided by citizens. |
| RQGA-8 | General Monitoring | The solution has to allow monitoring the exchanged messages as well as visualizing the results of validating them according to the law. |

Public Agencies Administrators have to be able to manage the aspects of the solution concerning their public agency. Table 2 lists and describes the specific requirements for this type of users.

**Table 2:** Requirements for Public Agencies Administrators

| ID | Name | Description |
|---|---|---|
| RQPA-1 | Consents Management | The solution has to allow managing the consents provided by citizens to the public agency. In addition, given a citizen and a purpose, the solution has to return the missing consents. This way, public agencies can request them to the citizen. |
| RQPA-2 | Personal Data Requests Management | The solution has to allow managing the life cycle (e.g. created, completed) of personal data requests performed by the citizens. |

Citizens have to be able to manage the consents they provide and the personal data requests they perform. Table 3 lists and describes the specific requirements for this type of users.

**Table 3:** Requirements for Citizens

| ID | Name | Description |
|---|---|---|
| RQCT-1 | Consents Management | The solution has to allow citizens to manage the consents they provide to the different public agencies. |
| RQCT-2 | Personal Data Requests Management | The solution has to allow citizens to perform personal data request to the different public agencies as well as visualizing the state of each request. |

Finally, Table 4 lists and describes the requirements associated with message exchanges.

**Table 4:** Requirements for Message Exchanges

| ID | Name | Description |
|---|---|---|
| RQME-1 | Validate Messages | The solution has to intercept all the messages that arrive at the platform and perform the required validations. |
| RQME-2 | Apply Actions | When a validation fails for a given message exchange, the solution has to apply the configured actions according to the operation that is being invoked. |

### 3.5 Final Remarks

It is important to note that although the analysis presented in this section was based in the Uruguayan context, it can also be applied in other countries. This is due to the fact that interoperability platforms are increasingly used in e-government scenarios [2][25] and that the countries where they are applied have promulgated some sort of data protection regulations [26], which are similar to the Uruguayan one (e.g. the organic law 15/1999 of Spain [27]).

## 4 Proposed Solution

This section presents the proposed solution which extends an interoperability platform to manage, monitor and enforce data protection regulations. More details concerning the solution can be found in [28].

First, the general architecture of the solution and its main components are presented. Then, the most relevant characteristics of the proposal are described. Finally, the key interactions between the components of the solution, in particular the one that takes place between its PDP and PEP modules, through XACML messages, are described.

### 4.1 General Architecture

As depicted in Figure 8 (a), the proposed solution extends an interoperability platform with a data protection component which is responsible for providing the functionalities to the three types of users identified in the previous section, as shown in Figure 8 (b). On one hand, this component allows general administrators to configure the required aspects for controlling messages exchanges through the platform as well as to monitor the solution. On the other hand, this component allows public agencies administrators to manage personal data requests performed by citizens. Finally, the extension allows citizens to manage the consents they provide as well as to perform personal data requests to the different public agencies.

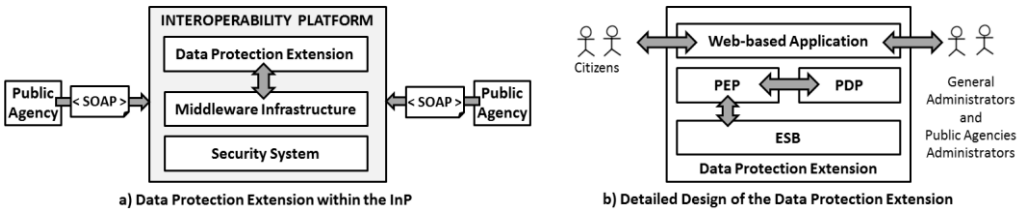


**Figure 8:** Logical Architecture

The main idea of the proposal is processing all the messages that pass through the interoperability platform. Each message is inspected by the solution in order to validate its content considering the aspects of the data protection law previously analysed. In particular, the validation considers the type of personal data that is being exchanged (i.e. public or sensitive) and the consents provided by citizens regarding these data. If the validation fails, the solution performs different pre-configured actions to the original message (e.g. execute a message transformation to take out elements which do not have the required consents to be shared).

In order to perform these tasks, the solution has to store a set of configuration data. For example, the solution stores the consents provided by the citizens, the configuration of Web Services and operations provided by public agencies and the actions to be taken for each operation if a validation fails when it is invoked.

As presented in Figure 8 (a), messages received by the interoperability platform are routed to the data protection component by the MI. Then, as shown in Figure 8 (b), within the data protection component, messages are first processed by an ESB which routes them to the PEP. The PEP component receives messages and sends XACML access requests to the PDP. The PDP component evaluates XACML requests, according to the available consents, and generates XACML responses which are returned to the PEP. Finally, messages are routed to the ESB which applies the required actions (e.g. a message transformation to take out elements) according to the XACML response.

The data protection component also includes a web-based application through which the different functionalities of the solution (e.g. consents management, personal data management) are delivered to the three types of users identified (i.e. general administrators, public agencies administrators and citizens).

### 4.2 Main Functionalities

This section presents the main functionalities of the proposed extension which address the requirements identified in Section 3.4.

#### *4.2.1 Personal Data Management*

Personal data management functionalities allow specifying and configuring the personal data that the platform is going to supervise. For example, administrators can specify that the platform is going to supervise the name, the last name and the address of citizens.

The solution follows the canonical data model pattern [20], given that public agencies can use different structures or names to represent citizens' personal data. For example, a public agency can use "name" to refer to the name of a citizen and other agency can use "first name". By using a Canonical Personal Data Model (CPDM), the platform manages a single data model which is mapped to the different elements (i.e. input and output parameters) of the Web Services' operations provided by the different public agencies.

In order to know which elements of an operation have to be validated within a message exchange, personal data have to be configured as public or sensitive. In particular, the solution allows classifying personal data, included in the CPDM, in one of the following three categories:

- Free: This type of personal data does not require the consent of citizens to be shared.
- Limited: This type of personal data requires the explicit consent of the data owner (i.e. a citizen) to be shared.
- Denied: This type of personal data cannot be shared.

Note that the functionalities described in this section address the requirement RQGA-3 (Table 1).

#### *4.2.2 Web Services Management*

Web Services management functionalities allow configuring Web Services and operations that public agencies provide through the platform. In particular, these functionalities allow general administrators to specify the Web Services that are provided by each public agency as well as the operations these Web Services offer. In addition, they allow specifying the mappings between the CPDM and the elements of Web Services operations. Figure 9 presents an example, where the elements of a message are mapped to the personal data in the CPDM.

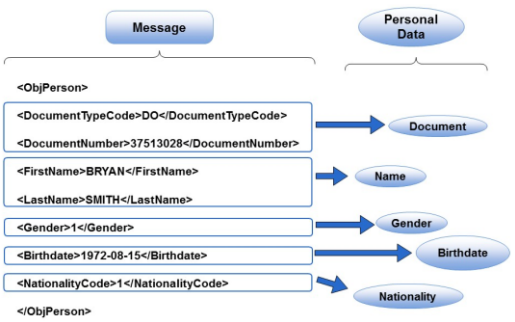


**Figure 9:** Mapping between Message Elements and Personal Data

Note that the functionalities described in this section address the requirement RQGA-6 (Table 1).

### *4.2.3 Consent Management*

Consent management functionalities allow citizens to manage the consents that they provide to public agencies through a web-based application. This way, citizens have more control over their personal data and can stop sharing them at any time. These functionalities also allow citizens to perform personal data requests to public agencies as well as monitoring the state of these requests. In addition, once these requests are completed by public agencies citizens are notified, for example, via email.

Consent management functionalities also provide utilities for public agency administrators. On one hand, these administrators can manage the personal data requests that citizens perform. On the other hand, they can obtain the missing consents to complete a purpose (e.g. an e-government procedure) for a given citizen. This way, public agency administrators can request these consents to the citizen to be able to complete a purpose.

Figure 10 presents an example in order to describe how the solution determines the missing consents given a purpose and a citizen.

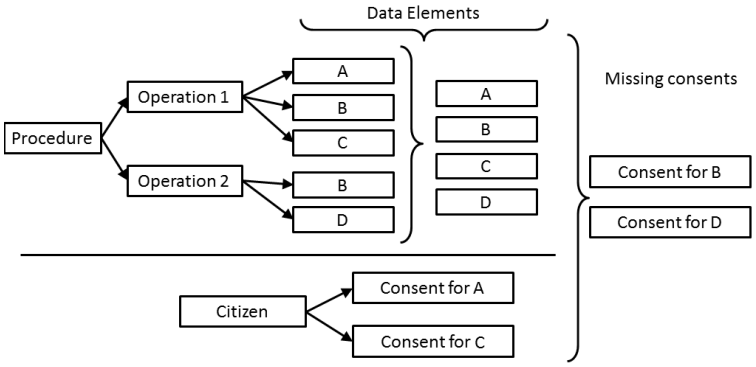


**Figure 10:** Determining Missing Consents

In particular, the figure shows a procedure which requires two operations to be complete: Operation 1 and Operation 2. Each operation uses a set of data elements: Operation 1 uses A, B and C while Operation 2 uses B and D. The union of these sets is a set with the data elements used by the procedure. On the other hand, a citizen has provided consents for two data elements: A and C. The missing consents for the citizen are calculated as the difference between the data elements used by the procedure (i.e. {A, B, C, D}) and the data elements for which the citizen has provided consents (i.e. {A, C}). This way, in order to complete the procedure for the given citizen, this citizen has to provide consents for the data elements B and D.

Note that the functionalities described in this section address the requirements RQPA-1 (Table 2) and RQCT-1 (Table 3).

### *4.2.4 Message Validation and Actions*

When a public agency sends a message to invoke a service, or to respond to a service request, the solution validates that all the data contained in the messages have the required consents to be exchanged. In particular, it validates that personal data of type “Limited” have the explicit consent of the data owner. If this validation fails, the solution enforces the data protection law by taking different actions. To this end, the solution provides configuration capabilities which allow specifying which actions have to be taken for each operation.

First, the solution allows specifying an XSLT transformation which will be applied to the message if the validation fails. For example, a transformation may take out some data elements of the message. For example, Figure 11 presents the results (on the right) of applying a transformation to take out two elements of the original message (on the left) sent to the platform given that the data owner did not provide consents to share some elements.

In addition, for each method the solution allows specifying a set of native actions which are also executed when a message validation fails. Some examples of actions are a notification via email and a notification via SMS. The extensible design of the solution allows including new types of actions.

Note that the functionalities described in this section address the requirements RQME-1 and RQME-2 (Table 4).

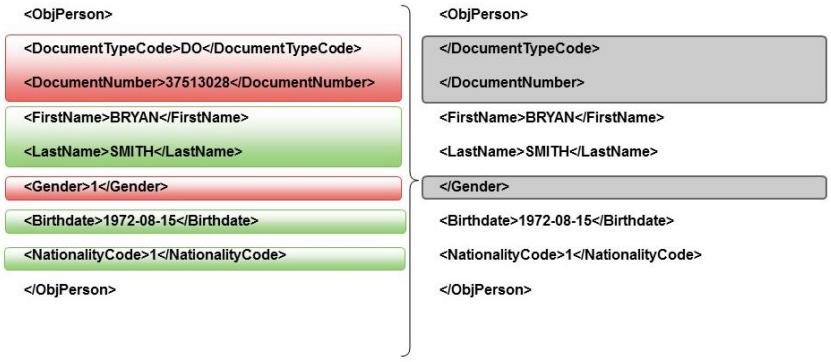


**Figure 11:** Filtering a Message

### 4.3 Components Design

The detailed design of the solution comprises five components: Data Protection Web (DP-Web), Data Protection Service (DP-Service), Data Protection Repository (DP-Repo), Data Protection Flow (DP-Flow) and Data Protection Service Flow (DP-ServiceFlow). Each component has a concrete responsibility aiming to achieve a loosely coupled solution. Figure 12 graphically presents these components as well as their dependencies.

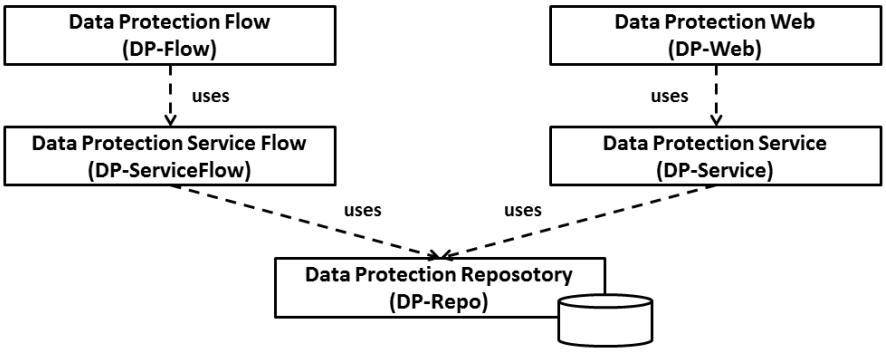


**Figure 12:** Components Diagram

DP-Web is a web-based application which can be used by administrators, to perform Configuration and Monitoring tasks, as well as by citizens, to manage their consents.

DP-Service is the component which contains the business logic used by the web application. It uses the DP-Repo component to manage the data required by the solution (e.g. configuration data, citizens' consents).

DP-Repo is the component which encapsulates the access to the database where all the data required by the solution are stored.

DP-ServiceFlow is the component which contains the logic to validate a message exchange. It has the PDP role in the solution given that it returns an authorization decision for the messages according to the available consents. It uses the DP-Repo component with the aim of obtaining the consents that are stored in the database.

In order to validate a message exchange, DP-ServiceFlow uses data coming from the DP-Flow component as well as configuration data. In particular, it uses the public agency which is sending the message, the service and operation that are being invoked, the public agency that provides that service, the citizen which owns the data that is being exchanged, the not expired consents provided by this citizen concerning those data, the purpose of the message exchange and the mappings between message elements and the CPDM. With these data, DP-ServiceFlow takes an authorization decision which is returned to the requester (i.e. DP-Flow).

Finally, DP-Flow is the component which has the PEP role in the solution. It is built on top of an ESB which allows leveraging its mediation capabilities. In particular, DP-Flow exposes an endpoint through which all the SOAP messages that arrive at the interoperability platform to invoke services are intercepted. These messages are processed by the DP-Flow component in order to obtain the required information to request an authorization decision to the DP-ServiceFlow component (i.e. the PDP).

In particular, DP-Flow uses the WS-Addressing headers to obtain the service and operation that are being invoked, the WS-Security headers to obtain the public agency that is performed the invocation and an additional header, defined by the proposed solution, to obtain the purpose of the invocation and an identification of the citizen that owns the data included in the message. Figure 13 presents the structure of this additional header.

```
<dp:dpInfo xmlns:dp='http://schemas.xml.dp'>
        <dp:purpose>T201</dp:purpose>
        <dp:document>4728398</dp:document>
        <dp:type_document>passport</dp:type_document>
</dp:dpInfo>
```

**Figure 13:** Additional Header

### 4.4 Components Interaction

Figure 14 presents a sequence diagram describing the interaction between some of the components presented in the previous section. In particular, the figure shows the interaction between the DP-Flow component and the DP-ServiceFlow component which, as stated before, have the PEP role and the PDP role in the solution, respectively.

Briefly, the main steps that are taken by the solution to process a message that arrives at the platform are:

1. When a message arrives at the platform to invoke a service, it is routed to the DP-Flow component (PEP) by the MI.
2. DP-Flow validates the structure of the message and, based on its content, builds an XACML request.
3. The XACML request is sent to the DP-ServiceFlow component (PDP) to obtain an access response.
4. The DP-ServiceFlow component gets the available consents from the DP-Repo component and, based on those, builds an XACML response.
5. The XACML response is sent to the DP-Flow component, which based on that response performs the required actions over the message (e.g. transformations).
6. Finally, the modified message is sent to the MI.

As explained before, the communication between DP-Flow and DP-ServiceFlow is performed through XACML messages. The proposed solution uses the elements of the XACML standard as follows.

XACML requests use the following XACML elements:

- Subject: It is the entity which requests access to a resource. In the proposed solution, this element holds a value which identifies the public agency that sends the SOAP message to invoke a service.
- Resource: It is the entity for which access is requested. In the proposed solution, the resource is the message that is sent as it contains personal data.
- Action: It is the action that is going to be performed over the resource. In the proposed solution, the action is always “Send” given that this is what public agencies need to do.
- Environment: It is a set of attributes which are relevant to take the authorization decision and do not depend on the Subject, Resource or Action. In the proposed solution this set includes the purpose of the message exchange, the identifier of the public agency to which the message is sent and the identification number of the citizen who owns the data that are sent in the message.

XACML responses use the following XACML elements:

- Decision: It is the authorization decision. The proposed solution uses the “Permit” and “Deny” values which are defined in the XACML standard.
- Status: It indicates if the evaluation of a decision request generates errors and it optionally specifies information about these errors.
- Obligations: It is a list of operations to be executed by the PEP for a given authorization decision. This section includes the actions to be taken (e.g. sending a notification via email) when some of the required consents to exchange a message are missing. When required, it also includes the identification of the transformation to be performed over the message. For example, this transformation could filter the personal data which are not authorized to be shared by the citizen.

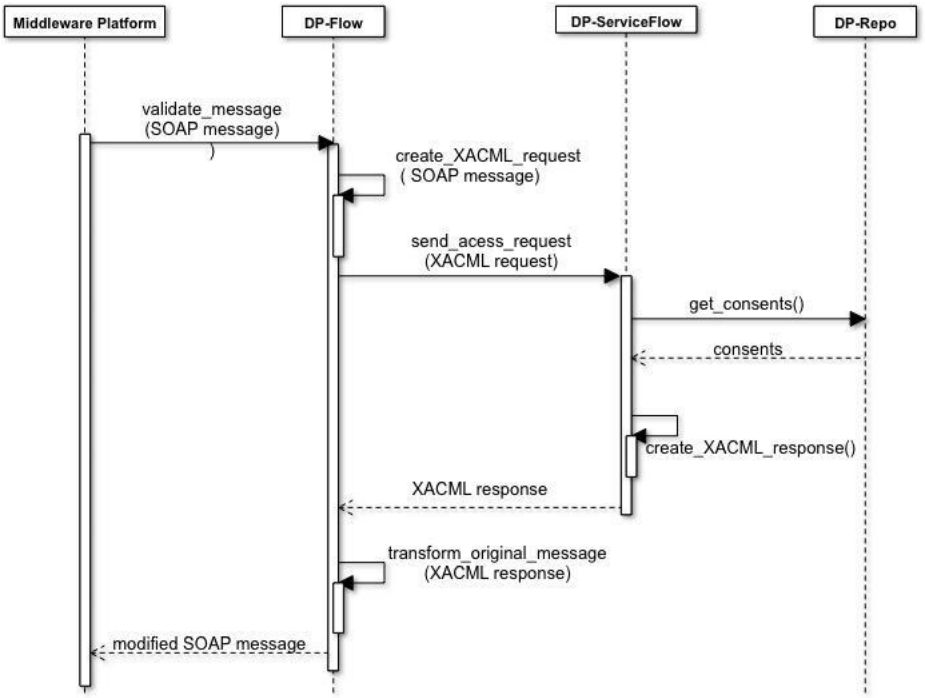


**Figure 14:** Sequence Diagram

## 5 Implementation and Experimentation

This section presents details concerning the implementation of the proposal as well as experimentation results.

### 5.1 Implementation Details

The solution proposed in Section 4 was completely prototyped and a demo is available on-line[4]. This section presents implementation details of this prototype including the software leveraged to implement it (Section 5.1.1), detailed aspects of some components (Section 5.1.2 and 5.1.3) and other implementation issues (Section 5.1.4).

#### *5.1.1 Base Software, Tools and Frameworks*

The prototype was implemented using Java Enterprise Edition 7[5] and deployed on the JBossEAP[6] platform. The implementation of the solution was mainly based in SwitchYard ESB[7]. In addition, the Spring Framework[8] was used in order to deal with different issues: life cycle management of objects, dependency injection and transaction management.

MySQL[9] was used as the database engine and the Hibernate[10] framework was leveraged for the object-relational mapping.

The Web-based application was developed using JSF[11] with bootstrap[12]. Also, Morris.js[13] was leveraged for the administration dashboard.

---

[4] http://www.fing.edu.uy/inco/grupos/lins/demos/demo-clouseau.mp4
[5] http://www.oracle.com/technetwork/java/javaee/
[6] http://www.jboss.org/products/eap/
[7] http://switchyard.jboss.org/
[8] http://projects.spring.io/spring-framework/
[9] https://www.mysql.com/
[10] http://hibernate.org/
[11] http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html
[12] https://github.com/twbs/bootstrap
[13] http://morrisjs.github.io/morris.js/

*5.1.2 DP-ServiceFlow Implementation*

The implementation of the DP-ServiceFlow component is based on the SwitchYard ESB. It exposes an endpoint through which all SOAP messages that arrive at the platform are intercepted. It acts as the PEP component of the solution given that it performs authorization requests and enforces the returned authorization decisions.

Figure 15 presents the detailed implementation of this component as a mediation flow of SwitchYard ESB, specified using Service Component Architecture (SCA)[14].

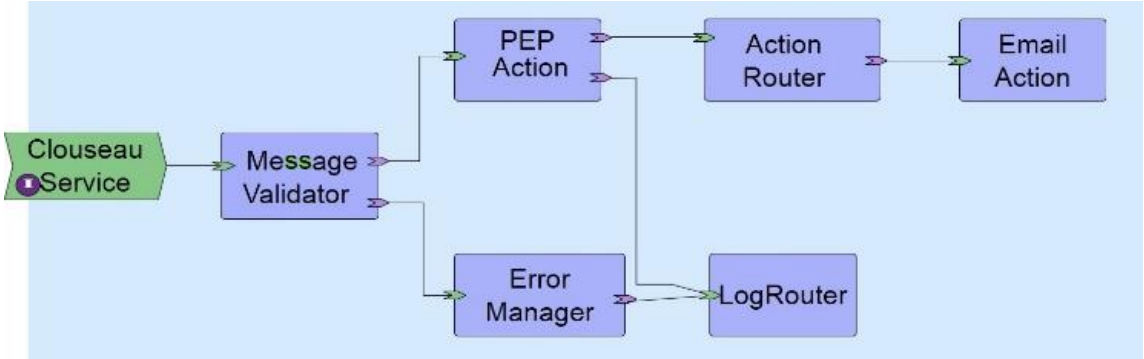


**Figure 15:** Detailed Implementation of DP-ServiceFlow

The Message Validator component has the responsibility to verify that messages are structurally valid, that is, they contain all the required data to be able to perform the consents validation. If messages are valid, they are routed to the PEP Action component; otherwise, they are routed to the Error Manager component.

The Error Manager component adds the current error to the message body and routes the message to the Log Router component.

The PEP Action component processes SOAP messages to extract the required data with the aim of building an XACML request to be sent to the DP-ServiceFlow component (i.e. the PDP). The XACML response returned by the PDP indicates if the involved public agencies are authorized to exchange the personal data included in the message. If not, the response also includes the action to be taken to enforce the law. If public agencies are authorized to exchange the personal data, the message is routed to the Log Router component; otherwise, it is routed to the Action Router component.

The Action Router component obtains from the XACML response the actions to be taken, including the XSLT transformation that should be applied to the message. In order to execute these actions, the solution leverages the Routing Slip pattern by adding to the message header the list of components through which it has to pass.

The Email Action component sends mails to general administrators each time a message arrives at the platform and missing consents are detected.

The Log Router component logs every intercepted message. In particular, for each message it stores: the public agency that sends the message, the public agency to which the message is sent, the name of the invoked operation, the original message that arrives at the platform and the modified message (i.e. after actions have been applied to it).

*5.1.3 Dashboard*

The solution provides a Dashboard which allows visualizing various metrics concerning the traffic and operation of the system. The Dashboard uses the data stored by the Log Router component in order to calculate the metrics.

Some examples of the information provided by the Dashboard are the number of messages (authorized and not authorized) in the last seven days (Figure 16), the number of messages per hour (Figure 17) and the list of the last messages that have arrived at the platform (Figure 18).

[14] http://www.oasis-opencsa.org/sca

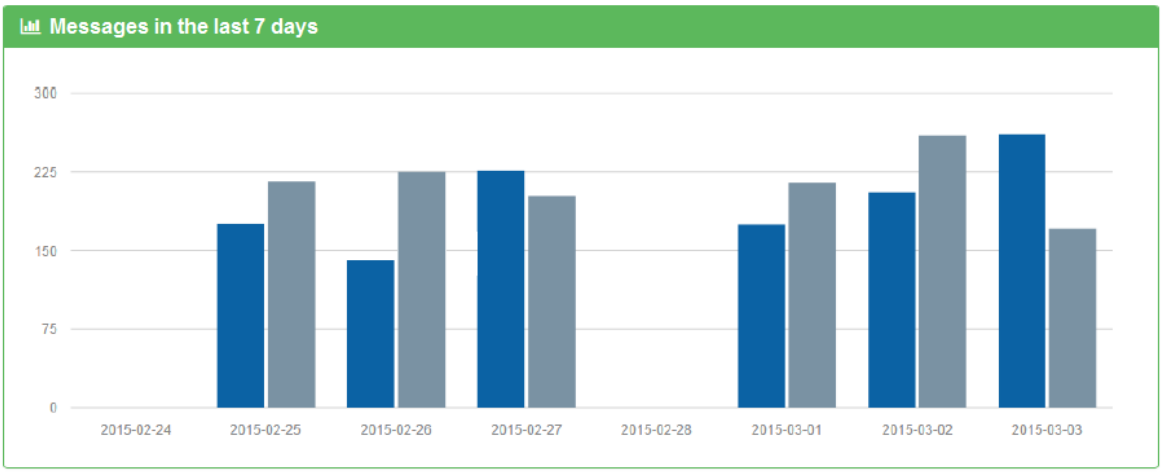


**Figure 16:** Messages in the Last Seven Days

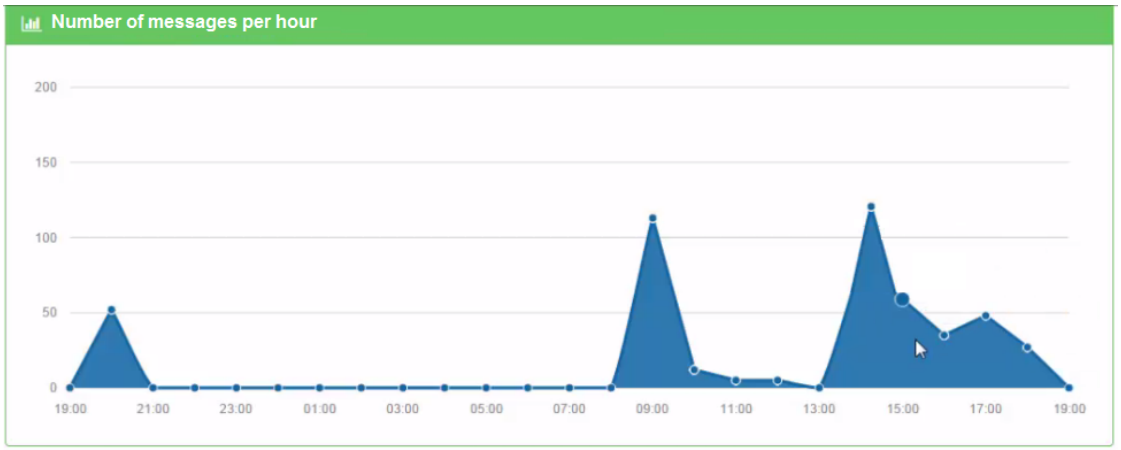


**Figure 17:** Number of Messages per Hour

**Last Messages**

| Date / Time | Origin Agency | Target Agency | Operation | Result |
|---|---|---|---|---|
| 03-03-2015 17:14 | MSP | DNIC | ObtPersonaPorDoc | AUTORIZED |
| 03-03-2015 17:13 | MSP | DNIC | ObtPersonaPorDoc | NOT AUTORIZED |
| 03-03-2015 17:13 | MSP | DNIC | ObtPersonaPorDoc | AUTORIZED |
| 03-03-2015 17:09 | MSP | DNIC | ObtPersonaPorDoc | NOT AUTORIZED |

**Figure 18:** Last Messages in the System

*5.1.4 Other Implementation Details*

With the aim of facilitating the configuration of new operations in the system as much as possible, the prototype processes WSDL files in order obtain the operations as well as their input and output parameters. To this end, the prototype leverages the javax.wsdl library. If the WSDL file cannot be processed (e.g. due to an invalid structure) the prototype also allows configuring operations in a manual way.

The prototype includes a native set of actions which can be taken to enforce data protection laws. In addition, it provides a mechanism through which new types of actions can be included in the system at runtime. To this end, the Reflection support provided by Java was leveraged. In particular, in order to create new types of actions a developer

have to implement a Java class, which should extends a generic Java class, and include it in the solution. At runtime, an instance of this class is created, using the Reflection support, and can be used as an action to enforce data protection regulations.

Finally, given that consents can expire, the prototype includes a Consents Remover which removes expired consents. In order to implement this remover, the scheduled task mechanism provided by Spring was used. This mechanism executes tasks every configurable time intervals using cron expressions.

### 5.2 Case Studies

This section presents two case studies which describe how the system operates in order to: i) validate the messages received by the platform for invoking an operation and ii) allow citizens perform personal data requests to public agencies.

#### *5.2.1 Messages Validation*

This case study describes the complete cycle of registering, configuring and invoking an operation in the system. In particular, the case study shows the exchanged messages to perform the invocation and how the system transforms them to enforce the data protection law.

The Web Service used in this case study is the "Basic Information Service"[15] provided by the National Direction of Civil Identification of Uruguay (DNIC). In particular, the case study focuses on an operation of this service called "ObtPersonaPorDoc" which given an identification number returns personal data of a citizen.

The registration of the operation in the system can be performed by uploading a WSDL description of the service[16] which is processed to automatically obtain the input and output parameters of the operation. In addition, an XSLT transformation has to be specified in case some data have to be filtered in a message exchange. Finally, the mappings between the parameters of the operation and the elements of the CPDM have to be configured. Table 5 presents the configuration of the operation "ObtPersonaPorDoc".

**Table 5:** Configuration of an Operation

| Message Element | Parameter Type | Personal Datum in the CPDM | Type of Personal Datum |
|---|---|---|---|
| TipoDocumento | INPUT | Document | LIMITED |
| NroDocumento | OUTPUT | Document | LIMITED |
| CodTipoDocumento | OUTPUT | Document | LIMITED |
| Nombre1 | OUTPUT | Name | FREE |
| Nombre2 | OUTPUT | Name | FREE |
| Apellido1 | OUTPUT | Name | FREE |
| Apellido2 | OUTPUT | Name | FREE |
| Sexo | OUTPUT | Gender | LIMITED |
| FechaNacimiento | OUTPUT | Birthdate | LIMITED |
| CodNacionalidad | OUTPUT | Nationality | DENIED |

This configuration data allows the solution to take actions when data elements try to be exchanged without the required authorization. For example, Figure 19 - a) presents the original response message of the "obtPersonaPorDoc" operation and Figure 19 - b) shows a message filtered by the solutions according to the configuration of the operation.

In particular, the values of the elements "CodTipoDocumento", "NroDocumento" and "Sexo" were taken out given that their type is LIMITED and the citizen did not provide consents to the DNIC to share them. In addition, the value of the element "CodNacionalidad" was taken out given that its type is "DENIED".

[15] http://www.agesic.gub.uy/innovaportal/v/1799/9/agesic/servicio-basico-de-informacion.html
[16] http://www.agesic.gub.uy/innovaportal/file/1799/1/servicio_basico_informacion.wsdl

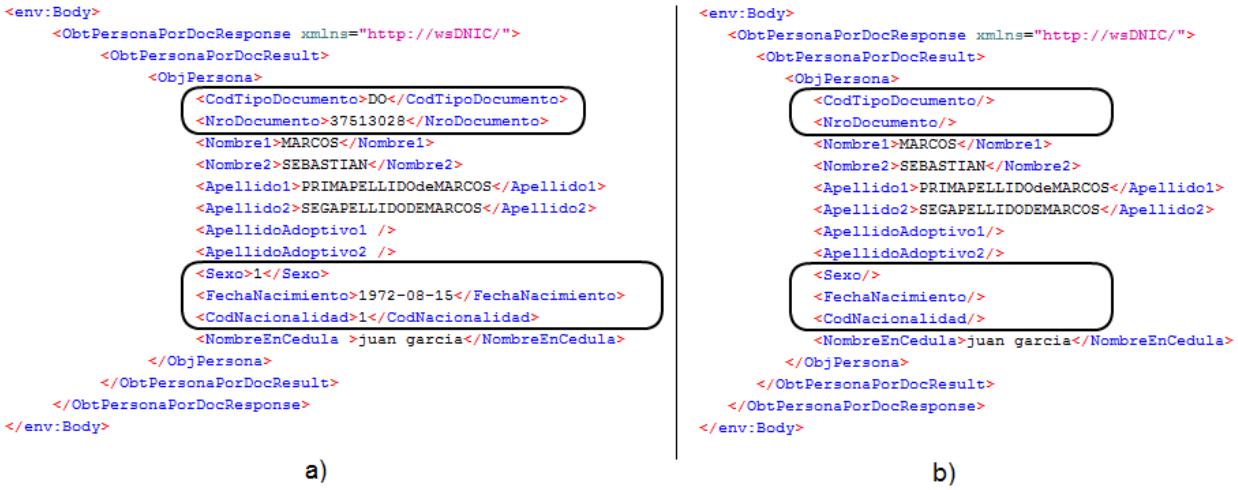


**Figure 19:** Message Responses: a) original service response, b) filtered service response

*5.2.2 Performing Personal Data Requests*

This case study presents how citizens manage personal data access request using the web-based application provided by the solution. In particular, Figure 20 presents how a citizen can create a personal data request specifying the public agency to which the request is going to be sent.


**Figure 20:** Creation of a personal data access request by a citizen

In addition, as shown in Figure 21, citizens can visualize a list of the requests performed by them, where the state of each request is indicated.


**Figure 21:** Personal Data Access Requests performed by a Citizen

### 5.3 Response Time Tests

With the goal of analyzing the overhead of applying different actions over the messages (e.g. transformations) when they do not have all the required consents to be exchanged, response time tests were performed. In particular, the processing time of invoking the service presented in Section 5.2.1 was obtained in two situations:

- All the required consents for invoking the service were available and the application of transformations was not required
- Consents for invoking the service were missing and a transformation had to be applied. In particular, a transformation that filters the not authorized elements was applied and a notification via email was sent to the administrator.

The tests were executed in a single notebook with an Intel(R) Core(TM) i3 processor (2.53 GHz) and 4GB of RAM. Table 6 presents the results of these tests.

**Table 6:** Response Times

| Test | Number of Invocations | Average Response Time (ms) |
|---|---|---|
| Without modifying the message | 500 | 153 |
| Filtering the message | 500 | 160 |

On one hand, these results show that the platform does not introduce a considerable overhead, given that the average processing time through the platform is 160 ms which is a reasonable response time for invoking a Web Service. On the other hand, the additional time required to apply actions to the message (i.e. filtering content and sending an email) is significantly lower than the total time required for processing the message.

### 5.4 Lessons Learned

The implementation of the proposal with specific products and the development of use cases allow us to identify some lessons which may be useful for governments which may decide to leverage this proposal.

Regarding implementation, even though the proposal presented in Section 4 is based on established integration patterns usually supported by ESB products (e.g. routing), the development effort for implementing the solution significantly depends on the selected ESB product. This is due to the fact that ESBs from different providers usually present dissimilar architectures and internal design. Consequently, governments which may want to leverage this solution should carefully choose the ESB product which better fits their specific needs regarding data protection.

In addition, the development of case studies let us realize that in order to leverage this solution, users with the administrator role should have a complete understanding of the data protection regulations as well as of the data exchanged in service invocations. This is required to be able to correctly configure all the aspects of the solution (e.g. which personal data are public or sensitive, which data element in a service invocation corresponds to personal data). Failing in configuring the system correctly may lead to the malfunction of the proposed solution.

As a more general reflection on these matters, the increasing demand for regulatory compliance, not only on data protection but also in very wide areas (e.g. finance, health, environmental), requires the definition of generic compliance mechanisms capable of dealing with different regulations and protocols.

## 6 Related Work

Monitoring and enforcing regulatory compliance requirements in large-scale software systems have been addressed by various authors in the last decade. One of the most relevant projects in this area is the European project COMPAS [29] that proposes an integrated solution for runtime compliance governance in SOA. In the context of this project various tools have been developed which allow modelling compliance requirements, linking them to business processes, monitoring process execution, displaying the current state of compliance and analysing cases of non-compliance [30][31]. This work is similar to ours given that it addresses regulatory compliance issues in large-scale software systems and it leverages sophisticated middleware technologies (e.g. ESB) to provide solutions. However, the main differences with our work are: i) the project mainly focuses on compliance requirements of business processes running under the supervision of a single organization while our work focuses on compliance requirements of inter-organizational interactions; ii) the project deal with general compliance requirements which are then refined in some specific cases (e.g. quality of service requirements, licensing requirements) but it does not provide solutions for compliance requirements related to data protection regulations; and iii) although performing corrective actions is mentioned [31], the project mainly deals with monitoring tasks while our work also addresses enforcement activities. Another relevant project in the area is the $C^3$ Pro project[17] which deals with compliance issues in cross-organizational business processes. In particular, the authors define a property of "compliability" [32][33] to characterize interaction models consistent with a set of compliance rules. This work is similar to our in that it focuses on inter-organizational interactions. However, compared to our work, the project focuses on design

[17] https://www.uni-ulm.de/en/in/dbis/research/projects/c3pro.html

time checking of compliance rules [34] while ours focuses on run-time checking mechanisms. In addition, this work does not deal either with data protection regulations and it does not provide runtime enforcement mechanisms.

Dealing with privacy related issues in e-government interoperability platforms has also been addressed in the literature. In [35] the authors present the development of a prototype which implements, using software agents, an e-government interoperability model that preserves privacy during the dynamic orchestration of services. Compared to our work, this proposal deals with privacy at a higher level (i.e. organizational level) given that it does not consider citizens' consents as specified in many data protection laws. Another difference with our approach is that the technological solution they propose is based on software agents and ours is based on middleware platforms. In addition, [36] presents the STORK project which aims to achieve the interoperability of the European electronic identifiers, in order to allow European citizens to establish new e-relations with the Public Administration and private sectors. Although this proposal argues that privacy is guaranteed and enforced by design, given that the platform does not store any personal data, the main focus of this work is interoperability of electronic identifiers across different European regions and countries. On the contrary, our work focuses on the specific requirements posed by data protection laws. Finally, in [37] the authors formalize data protection requirements generated by the German regulation and prototypes a semi-automatic tool to help service providers to verify that their services comply with this regulation. Compared to our proposal, this work does not consider citizens' consents and it does not leverage the mechanisms provided by middleware platforms in order to implement the proposed solutions.

In addition, several works have addressed the issues of data protection in the Health area. For example, in [38], the authors propose an event-based architecture to enforce privacy regulations in an inter-organisational scenario involving social welfare and health systems. In turn, requirements on Information Systems enabling to enforce Health care regulations are identified in [39]. The main difference of these works with our proposal is that they do not handle users' consents to monitor and enforce data protection regulations. Finally, [40][41] consider users' consents but the solutions proposed in these papers are presented at a higher level of abstraction compared to our proposal which presents in a detailed way how citizens' consents are managed and enforced through an ESB.

Concerning cloud-based proposals, some methods for monitoring and enforcing privacy regulations in cloud platforms have been proposed. In [42] the privacy enforcement mechanism leverages Aspect Oriented Programming (AOP) features to add privacy-related meta-information to business applications by using the Java annotation mechanism. The proposal also includes components to filter database operations through JDBC and SQL interceptors. Compared to our approach, this work does not deal with data protection within inter-organizational data exchanges. Finally in [43], the authors describe the design of a framework for automating the collection of evidence regarding obligations concerning personal data transfers. However, this work only focuses on monitoring aspects while ours also considers enforcement mechanisms.

Considering the limitations of this related work, our proposal focus on providing solutions for the concrete compliance requirements posed by data protection laws [12][28][44] as part of a general compliance-aware inter-organizational integration platform [45][46]. The main differences of the latter with the proposals mentioned before are: (i) the latter's approach performs compliance monitoring and enforcement in the middleware connecting business application, which is an integration platform; and (ii) it addresses the general issues of regulatory compliance including, but not limited to, data protection regulations.

## 7 Conclusions and Future Work

This paper proposed solutions to manage, monitor and enforce regulatory compliance related to Data Protection in e-government by using capabilities of interoperability platforms and minimizing the impact on business applications.

The main contributions of this work are: (i) defining and showing the feasibility of a platform-centric approach for data protection compliance enforcement, (ii) the analysis of the involved issues and the identification of a set of requirements, (iii) the solution proposal based on recognized integration technologies and security standards, and (iv) the solution implementation using an ESB product which enabled to validate the proposal.

Although the analysis and the solution were proposed in the Uruguayan context (AGESIC's e-government platforms and Uruguayan Data Protection laws) the proposed mechanisms may be applied in other similar contexts. This work constitutes a step forward on addressing the issues of defining strategies for efficiently monitor and enforce regulatory compliance using e-government interoperability platforms particularly on Data Protection regulations.

The implementation also led to interesting conclusions concerning the used standards and technologies. Firstly, the XACML standard responded to the requirements of enforcing Data Protection laws. Second, it was shown that ESBs provide the means to implement compliance enforcement actions on Data Protection regulations (e.g. through transformations). Particularly, the product SwitchYard has a big potential and natively solves many of the arisen problems. Nevertheless, the difficulties appeared in the implementation have shown aspects for improvement.

As future work, an extension to this work may consist in using a Complex Event Processing engine (CEP) in the proposed solution. This integration would allow generating alarms associated to certain events detected on real-time, for example, that a certain number of messages are blocked for the same organization. In addition, CEP could be used to enforce temporal conditions of Data Protection laws (e.g. as established by the Uruguayan regulations, a citizen has the right to receive the information about him managed by an organization with a delay of five working days after the request).

Finally, other future work would consist of: (i) analyzing the usage of the Privacy Policy Profile of XACML v3.0 to manage the concept of "purpose", (ii) analyzing and proposing solutions for scenarios where advanced Web Services standards are used (e.g. WS-Security for signing and encrypting SOAP messages) as this could hinder message transformations, and (iii) further analyzing and evaluating the performance and scalability of this type of solution.

**Acknowledgements**

This work was partially funded by the Comisión Sectorial de Investigación Científica (CSIC), Universidad de la República, Uruguay.

**References**

[1] John Akeroyd, "Information Architecture and e-Government," *INFuture2009:"Digital Resources and Knowledge Sharing*, pp. 687–701, 2009.

[2] Roberto Baldoni, Stefano Fuligni, Massimo Mecella, and Francesco Tortorelli, "The Italian e-Government Enterprise Architecture: A Comprehensive Introduction with Focus on the SLA Issue," in *Service Availability*, T. Nanya, F. Maruyama, A. Pataricza, and M. Malek, Eds. Springer Berlin Heidelberg, 2008, pp. 1–12.

[3] L. González, R. Ruggia, J. Abin, G. Llambías, R. Sosa, B. Rienzi, D. Bello, and F. Alvarez, "A Service-oriented Integration Platform to Support a Joined-up E-government Approach: The Uruguayan Experience," in *Proceedings of the Joint International Conference on Electronic Government, the Information Systems Perspective, and Electronic Democracy*, Vienna, Austria, 2012.

[4] M. Papazoglou and W.-J. Heuvel, "Service oriented architectures: approaches, technologies and research issues," *The VLDB Journal*, vol. 16, no. 3, pp. 389–415, 2007.

[5] Yuehua Wu, "Protecting personal data in E-government: A cross-country study," *Government Information Quarterly*, vol. 31, no. 1, pp. 150–159, Jan. 2014.

[6] Parlamento Uruguayo, "Ley Nº 18.331 - Protección de Datos Personales y Acción de "Habeas Data"", http://www.parlamento.gub.uy/leyes/AccesoTextoLey.asp?Ley=18331, 2008.

[7] L. González and R. Ruggia, "Adaptive ESB Infrastructure for Service Based Systems," in *Adaptive Web Services for Modular and Reusable Software Development: Tactics and Solutions*, IGI Global, 2012.

[8] L. González and R. Ruggia, "Addressing QoS issues in service based systems through an adaptive ESB infrastructure," in *Proceedings of the 6th Workshop on Middleware for Service Oriented Computing - MW4SOC '11*, Lisbon, Portugal, 2011, pp. 1–7.

[9] L. González, "Plataforma ESB Adaptativa para Sistemas Basados en Servicios," Tesis de Maestría en Informática, PEDECIBA Informática | Instituto de Computación – Facultad de Ingeniería – Universidad de la República, 2011.

[10] D. Chappell, *Enterprise Service Bus: Theory in Practice*. O'Reilly Media, 2004.

[11] OASIS, "eXtensible Access Control Markup Language (XACML)" Version 3.0", http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html", 2013.

[12] A. Echevarria, D. Morales, and L. Gonzalez, "Monitoring and enforcing data protection laws within an e-government interoperability platform," in *2015 Latin American Computing Conference*, 2015, pp. 1–12.

[13] W3C, "Web Services Description Requirements", http://www.w3.org/TR/ws-desc-reqs/#definitions, 2002.

[14] W3C, "SOAP Version 1.2", https://www.w3.org/TR/soap12/, 2007.

[15] W3C, "WSDL Version 1.1", http://www.w3.org/TR/wsdl, 2001.

[16] W3C, "Web Services Addressing 1.0 – Core", https://www.w3.org/TR/ws-addr-core/, 2006.

[17] OASIS, “WS-Security 1.1”, https://www.oasis-open.org/committees/wss/, 2004.

[18] W3C, “XSL Transformations (XSLT) 1.0”, http://www.w3.org/TR/xslt, 1999.

[19] “UN E-Government Survey 2014”. http://unpan3.un.org/egovkb/en-us/Reports/UN-E-Government-Survey-2014, 2014.

[20] G. Hohpe and B. Woolf, Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional, 2003. http://www.eaipatterns.com/CanonicalDataModel.html.

[21] D. Doneda and L. S. Mendes, “Data Protection in Brazil: New Developments and Current Challenges,” in Reloading Data Protection, S. Gutwirth, R. Leenes, and P. D. Hert, Eds. Springer Netherlands, 2014, pp. 3–20.

[22] OCDE Privacy Framework, http://www.oecd.org/internet/ieconomy/privacy-guidelines.htm, 2013.

[23] “Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data”, Official Journal L 281 , 23/11/1995 P. 0031 - 0050. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.1995.281.01.0031.01.ENG, 1995.

[24] R. Del Villar, A. D. de Leon, and J. G. Hubert, “Regulation of Personal Data Protection and of Reporting Agencies: a Comparison of Selected Countries of Latin America, the United States and European Union Countries,” *Credit Reporting Systems and the International Economy, MIT Press*, 2001.

[25] Gobierno de Canarias, “Plataforma de Interoperabilidad del Gobierno de Canarias”, http://www.gobiernodecanarias.org/platino/.

[26] Unidad Reguladora y de Control de Datos Personales, Leyes Internacionales de Protección de Datos Personales, http://datospersonales.gub.uy/inicio/normativa/internacional/.

[27] Agencia Española de Protección de Datos, Agencia Española de Protección de Datos - Estatal, http://www.agpd.es/portalwebAGPD/canaldocumentacion/legislacion/estatal/index-ides-idphp.php.

[28] A. Echevarria and D. Morales, “Protección de Datos Personales en Plataformas de Integración,” Tesis de Grado, Universidad de la República, 2014.

[29] COMPAS, “COMPAS Project - Final Report”, http://cordis.europa.eu/fp7/ict/ssai/docs/finalreport-compas.pdf, 2011.

[30] A. Birukou, V. D’Andrea, F. Leymann, J. Serafinski, P. Silveira, S. Strauch, and M. Tluczek, “An Integrated Solution for Runtime Compliance Governance in SOA,” in *Service-Oriented Computing*, P. P. Maglio, M. Weske, J. Yang, and M. Fantinato, Eds. Springer Berlin Heidelberg, 2010, pp. 122–136.

[31] T. Holmes, E. Mulo, U. Zdun, and S. Dustdar, “Model-aware Monitoring of SOAs for Compliance,” in *Service Engineering*, Springer Vienna, 2011, pp. 117–136.

[32] D. Knuplesch, M. Reichert, J. Mangler, S. Rinderle-Ma, and W. Fdhila, “Towards Compliance of Cross-Organizational Processes and Their Changes,” in *Business Process Management Workshops*, M. L. Rosa and P. Soffer, Eds. Springer Berlin Heidelberg, 2013, pp. 649–661.

[33] D. Knuplesch, M. Reichert, W. Fdhila, and S. Rinderle-Ma, “On Enabling Compliance of Cross-Organizational Business Processes,” in *Business Process Management*, F. Daniel, J. Wang, and B. Weber, Eds. Springer Berlin Heidelberg, 2013, pp. 146–154.

[34] D. Knuplesch, M. Reichert, and A. Kumar, “Visually Monitoring Multiple Perspectives of Business Process Compliance,” in *Business Process Management*, H. R. Motahari-Nezhad, J. Recker, and M. Weidlich, Eds. Springer International Publishing, 2015, pp. 263–279.

[35] F. Marques, G. P. Dias, and A. Zúquete, “Agent-based interoperability for e-government,” Advances in Intelligent Systems and Computing, vol. 217, pp. 561–568, 2013.

[36] J. L. Hernandez-Ardieta, J. Heppe, and J. F. Carvajal-Vion, “STORK: The European Electronic Identity Interoperability Platform,” *IEEE Latin America Transactions*, vol. 8, no. 2, pp. 190–193, Apr. 2010.

[37] Christian Sillaber and Ruth Breu, “Managing legal compliance through security requirements across service provider chains: A case study on the German Federal Data Protection Act.,” in *GI-Jahrestagung*, 2012, pp. 1306–1317.

[38] G. Armellin, D. Betti, F. Casati, A. Chiasera, G. Martinez, and J. Stevovic, "Privacy Preserving Event Driven Integration for Interoperating Social and Health Systems," in *Secure Data Management*, W. Jonker and M. Petković, Eds. Springer Berlin Heidelberg, 2010, pp. 54–69.

[39] A. Siena, G. Armellin, G. Mameli, J. Mylopoulos, A. Perini, and A. Susi, "Establishing Regulatory Compliance for Information System Requirements: An Experience Report from the Health Care Domain," in Conceptual Modeling – ER 2010, J. Parsons, M. Saeki, P. Shoval, C. Woo, and Y. Wand, Eds. Springer Berlin Heidelberg, 2010, pp. 90–103.

[40] M. S. Shin, H. S. Jeon, Y. W. Ju, B. J. Lee, S.-P. Jeong, M. S. Shin, H. S. Jeon, Y. W. Ju, B. J. Lee, and S.-P. Jeong, "Constructing RBAC Based Security Model in u-Healthcare Service Platform," *The Scientific World Journal, The Scientific World Journal*, vol. 2015, 2015, p. e937914, Jan. 2015.

[41] M. Ulieru and D. Ionescu, "Privacy and security shield for health information systems (e-Health)," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences, 2002. HICSS*, 2002, pp. 496–501 Vol. 1.

[42] P. Yu, J. Sendor, G. Serme, and A. S. de Oliveira, "Automating Privacy Enforcement in Cloud Platforms" in Data Privacy Management and Autonomous Spontaneous Security, R. D. Pietro, J. Herranz, E. Damiani, and R. State, Eds. Springer Berlin Heidelberg, 2013, pp. 160–173.

[43] A. S. De Oliveira, J. Sendor, A. Garaga, and K. Jenatton, "Monitoring Personal Data Transfers in the Cloud" in 2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), 2013, vol. 1, pp. 347–354.

[44] F. Piedrabuena, L. González, and R. Ruggia, "Enforcing Data Protection Regulations within e-Government Master Data Management Systems," in 17th International Conference on Enterprise Information Systems, Barcelona, Spain, 2015.

[45] L. González and R. Ruggia, "Towards a Compliance-Aware Inter-organizational Service Integration Platform," in On the Move to Meaningful Internet Systems: OTM 2014 Workshops, 2014, pp. 8–17.

[46] L. González and R. Ruggia, "A reference architecture for integration platforms supporting cross-organizational collaboration," in *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services, iiWAS 2015, Brussels, Belgium, December 11-13, 2015*, 2015, p. 92.

## Appendix A – List of Abbreviations

AGESIC – Agencia de Gobierno Electrónico y Sociedad de la Información (Electronic Government and Information Society Agency)
CEP – Complex Event Processing
CPDM – Canonical Personal Data Model
CSIC – Comisión Sectorial de Investigación Científica
DNIC – Dirección Nacional de Identificación Civil (National Direction of Civil Identification)
DP-Flow – Data Protection Flow
DP-Repo – Data Protection Repository
DP-Service – Data Protection Service
DP-ServiceFlow – Data Protection Service Flow
DP-Web – Data Protection Web
EGP – E-Government Platform
EIP – Enterprise Integration Patterns
ESB – Enterprise Service Bus
InP – Interoperability Platform
MDM – Master Data Management
MI – Middleware Infrastructure
PAP – Policy Administration Point
PDP – Policy Decision Point
PEP – Policy Enforcement Point
PIP – Policy Information Point
SAML – Security Assertion Markup Language
SCA – Service Component Architecture
SOA – Service Oriented Architecture
SOAP – Simple Object Access Protocol
SS – Security System
WS-Addressing – Web Services Addressing
WS-Security – Web Services Security
WSDL – Web Services Description Language
XACML – eXtensible Access Control Markup Language
XSLT – eXtensible Stylesheet Language Transformations