Application for JPEG privacy implementation with XACML

A Degree Thesis
Submitted to the Faculty of the

Escola Tècnica Superior d’Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by
Miguel Castillo Nolasco

In partial fulfilment
of the requirements for the degree in
Audiovisual Systems Engineering

Advisor: Jaime M. Delgado Merce

Barcelona, February 2017
Abstract

This project is based on the development of an application for the management of privacy and security in JPEG images through policies designed in standard XACML.

The privacy policies are incorporated into the metadata defined for this type of data in the JPSearch standard for the compaction of the information in a single resource.

Dynamic queries (request) are generated with the purpose of obtaining permission to access an image by checking the privacy policies, where it is determined who, how and when can access. The evaluation of the request against the policy is carried out by Balana, and this return an answer (response) with the decision.

There are three independent lines of work, firstly, the design and creation of privacy policies based on the XACML standard, secondly, the update of JPSearch metadata in the corresponding fields for privacy and security and, as a last part, the definition of the system for the validation of a request to an image through the Balana implementation that incorporates the architecture defined in XACML.
Resum

Aquest projecte mostra el desenvolupament d'una aplicació per a la gestió de la privacitat i seguretat en imatges JPEG a partir de polítiques dissenyades en l'estàndard XACML.

Les polítiques de privacitat estan incorporades a les metadades definides per aquest tipus de dades a l'estàndard JPSearch per a la compactació de la informació en un sol recurs.

S'han generat consultes dinàmiques (request) amb l'intenció d'obtenir el permís d'accés a una imatge comprovant les polítiques de privacitat, on es determina qui, com i quan pot accedir. L'avaluació de la request contra la política la realitza Balana, i aquest ens torna una resposta (response) amb la decisió.

Hi han tres línies de treball independents, en primer lloc, el disseny i creació de polítiques de privacitat a partir del estàndard XACML, en segon lloc, l'actualització de les metadades de JPSearch en els camps corresponents per a la privacitat i seguretat i com a última part la definició del sistema per a la validació d'una petició a una imatge a través de l' implementació Balana que incorpora l'arquitectura definida en l'estàndard XACML.
Resumen

Este proyecto muestra el desarrollo de una aplicación para la gestión de la privacidad y seguridad en imágenes JPEG a partir de políticas diseñadas en el estándar XACML.

Las políticas de privacidad son incorporadas en los metadatos definidos para este tipo de datos en el estándar JPSearch para la compactación de la información en un solo recurso.

Se generan consultas dinámicas (request) con la intención de conseguir el permiso de acceso a una imagen comprobando las políticas de privacidad, donde se determina quién, cómo y cuándo puede acceder. La evaluación de la request contra la política lo realiza Balana, y este nos devuelve una respuesta (response) con la decisión.

Hay tres líneas de trabajo independientes, en primer lugar, el diseño y creación de políticas de privacidad a partir del estándar XACML, en segundo lugar, la actualización de los metadatos de JPSearch en los campos correspondientes para la privacidad y seguridad y como última parte la definición del sistema para la validación de una petición a una imagen a través de la implementación Balana que incorpora la arquitectura definida en XACML.
Acknowledgements

I want to thank my director Jaime Delgado for giving me the opportunity to carry out this project. He has always provided me with the help and advice to overcome stages of difficulty during the project.

I also want to thank my family, friends and Laura for the mental support during these years of career.
## Revision history and approval record

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>23/12/2017</td>
<td>Document creation</td>
</tr>
<tr>
<td>1</td>
<td>05/01/2018</td>
<td>Document revision</td>
</tr>
<tr>
<td>2</td>
<td>22/01/2018</td>
<td>Document revision</td>
</tr>
</tbody>
</table>

## DOCUMENT DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>Name</th>
<th>e-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miguel Castillo</td>
<td><a href="mailto:mcastillonolasco@gmail.com">mcastillonolasco@gmail.com</a></td>
</tr>
<tr>
<td>Jaime Delgado</td>
<td><a href="mailto:jaime.delgado@ac.upc.edu">jaime.delgado@ac.upc.edu</a></td>
</tr>
</tbody>
</table>

## Written by: Reviewed and approved by:

<table>
<thead>
<tr>
<th>Date</th>
<th>Name</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>23/12/2017</td>
<td>Miguel Castillo</td>
<td>Project Author</td>
</tr>
<tr>
<td>22/01/2018</td>
<td>Jaime Delgado</td>
<td>Project Supervisor</td>
</tr>
</tbody>
</table>
Table of contents

Abstract ..........................................................................................................................1
Resum ..........................................................................................................................2
Resumen .......................................................................................................................3
Acknowledgements ......................................................................................................4
Revision history and approval record ..........................................................................5
Table of contents ..........................................................................................................6
List of Figures ..............................................................................................................8
List of Tables ................................................................................................................9
1. Introduction ..............................................................................................................10
   1.1. Statement of purpose .......................................................................................10
   1.2. Requirements and specifications .................................................................10
   1.3. Methods and procedures ..............................................................................11
   1.4. Work plan .......................................................................................................11
       1.4.1. Tasks .....................................................................................................11
       1.4.2. Milestones ............................................................................................13
       1.4.3. Gantt Diagram ......................................................................................13
   1.5. Deviations and incidences ..............................................................................14
2. State of the art ..........................................................................................................14
   2.1. XACML ..........................................................................................................14
   2.2. Balana ...........................................................................................................15
   2.3. Encryption and Decryption ..........................................................................15
   2.4. XML Signature ..............................................................................................15
   2.5. WSO2 Identity Server ....................................................................................16
3. Methodology / project development ......................................................................16
   3.1. Creating Policies ............................................................................................16
       3.1.1. XACML .................................................................................................18
       3.1.2. WSO2 Identity Server ..........................................................................21
   3.2. Updating Metadata .........................................................................................25
       3.2.1. JPSearch ...............................................................................................25
       3.2.2. JPEGMetadataEditor ..........................................................................25
       3.2.2.1. Insert policy .....................................................................................26
       3.2.2.2. Insert XML signature .......................................................................27
       3.2.2.3. Encryption .......................................................................................27
3.3. Privacy System................................................................................................................. 28
  3.3.1. File Storage......................................................................................................................... 29
  3.3.2. Databases MySQL .............................................................................................................. 29
3.4. Core Structure Rights Validator App...................................................................................... 29
  3.4.1. Login and Image Select......................................................................................................... 29
  3.4.2. Extract Metadata.................................................................................................................. 30
  3.4.3. Obtain Policy and Signature ................................................................................................. 30
  3.4.4. Decryption ............................................................................................................................. 31
  3.4.5. Validate XML Digital Signature .......................................................................................... 31
  3.4.6. Generate Request ................................................................................................................ 31
  3.4.7. Balana ................................................................................................................................. 32
  3.4.8. Display and Updates ............................................................................................................ 33
4. Results ........................................................................................................................................ 34
  4.1. Privacy policy design .............................................................................................................. 34
  4.2. Rights Validator App .............................................................................................................. 34
  4.3. Test 1 ..................................................................................................................................... 35
  4.4. Another environment defined in the system ............................................................................. 37
  4.5. Test 2 ..................................................................................................................................... 38
5. Budget ........................................................................................................................................ 40
6. Conclusions and future development .......................................................................................... 40
Bibliography .................................................................................................................................... 42
Appendix A ...................................................................................................................................... 43
Appendix B ...................................................................................................................................... 45
  Policy Image 1 ................................................................................................................................. 45
  Policy Image 2 ................................................................................................................................. 46
  Policy Image 3 ................................................................................................................................. 47
  Policy Image 4 ................................................................................................................................. 48
Appendix C ...................................................................................................................................... 50
  C.1 Example A ............................................................................................................................... 50
  C.2 Example B ............................................................................................................................... 56
List of Figures

1.4.3 - Gantt Diagram ........................................................................................................ 13
2.1 - XACML Architecture ................................................................................................. 14
3.2 - Policies Structure........................................................................................................ 17
3.2.2.A - Starting WSO2 server ...................................................................................... 22
3.2.2.B - WSO2 Policy Editors ....................................................................................... 22
3.2.2.C - WSO2 Standard Policy Editor .......................................................................... 22
3.2.2.D - Description Policy Editor .................................................................................. 23
3.2.2.E - Target Policy Editor ......................................................................................... 23
3.2.2.F - Conditions Policy Editor ................................................................................... 23
3.2.2.G - Rules Policy Editor ........................................................................................... 24
3.2.2.H - Configure Rules Policy Editor ........................................................................... 24
3.2.2.I - Add Attributes Policy Editor ............................................................................. 24
3.3.1 - JPSearch Segment .............................................................................................. 25
3.3.2 - JPEGMetadataEditor .......................................................................................... 26
3.3.2.3 - Encryption Schema ......................................................................................... 28
3.3 - System Global Structure .......................................................................................... 28
3.4 - Core Structure RightsValidator ................................................................................ 29
3.4.1.A - Login ................................................................................................................ 29
3.4.1.B - Image select ...................................................................................................... 30
3.4.7 - XACML standard architecture ............................................................................ 32
3.4.8.A - Deny display .................................................................................................... 33
3.4.8.B - Permit display .................................................................................................. 33
4.4.1 - Schema of environments ..................................................................................... 37
4.4.2 - New login page ..................................................................................................... 38
4.4.3 - Images of new environment ................................................................................ 38
List of Tables

3.2.1 - Defined Policy ........................................................................................................21
4.3.1 - Users Company database ......................................................................................35
4.3.2 - Images Company database .....................................................................................35
4.3.3 - Test Summary Company Table ...............................................................................36
4.3.4 - Images Updated Company database ......................................................................36
4.3.5 - Users Updated Company database .........................................................................36
4.5.1 - Users Photography Database ..................................................................................39
4.5.2 - Images Photography Database ..............................................................................39
4.5.3 - Test Summary Photography Table ...........................................................................39
5 - Budget .........................................................................................................................40
6.A - Policy Configuration Table A ....................................................................................41
6.B - Policy Configuration Table B ....................................................................................41
1. **Introduction**

1.1. **Statement of purpose**

Currently, a lot of different JPEG images are shared due to the ease of creating them, using a camera taking a picture or even creating a file on a computer. The images travel through networks with great ease due to current technology. Privacy and security in these cases is non-existent, which is why it motivates the management of these areas.

This project consists of the development of an application for the privacy of JPEG images through security policies based on XACML language. The objective is a functional system that implements these technologies and demonstrates how the cases of privacy in images can be solved.

The project main goals are:

- Functional system with privacy policies in images JPEG.
- Evaluate access requests according to the rules defined in policies.
- Use of the XACML language for the creation of policies and for sending the requests.
- Integrate the policies in the metadata of the JPEG images.
- Establish practical examples for the use of the system
- Provide the system with maximum security possible with encryption and digital signature

1.2. **Requirements and specifications**

Project requirements:

1. Authorize or deny access according to the policy of the image
2. Encryption of the privacy policy
3. Inclusion of the digital signature
4. Variety of privacy policies that can be tested
5. Environment of access to images for a registered user

Project specifications:

1. Database with the information of users and images
2. Local or cloud storage for images
3. Use of the XACML protocol
4. Application designed with IDE Eclipse
5. Inclusion of the policies in the metadata of the image with the program "JPEGMetadataEditor"
6. Use Balana open source implementation based on XACML standard.
1.3. **Methods and procedures**

Based on previous projects, the first one "Metadata Interoperability with JPSearch" of the student Nicos Demetriou to modify metadata of a image. Later this application was modified for its adaptation to privacy metadata in the project "Privacy in images jpg by XACML" of the student Alberto Durán Montoro. This project finally generates a policy and authorization in a simple case in a desktop application with a static request.

This project consists of going a step further and completing an access system through the standard (XACML). In short, a user can verify that the system really works for the privacy of the images.

The XACML standard, the Balana implementation and the Eclipse IDE will be used to build the described application in Java.

1.4. **Work plan**

1.4.1. **Tasks**

<table>
<thead>
<tr>
<th>Project: Application for JPEG privacy implementation with XACML</th>
<th>WP ref: (WP#) 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major constituent: Search information</td>
<td>Sheet 1 of 5</td>
</tr>
<tr>
<td>Short description: Search information about the technologies.</td>
<td>Planned start date: 01-09-17</td>
</tr>
<tr>
<td></td>
<td>Planned end date: 01-10-17</td>
</tr>
<tr>
<td></td>
<td>Start event: 01-09-17</td>
</tr>
<tr>
<td></td>
<td>End event: 07-10-17</td>
</tr>
<tr>
<td>Internal task T1: Search information: XACML and Balana</td>
<td>Deliverables: None</td>
</tr>
<tr>
<td>Internal task T2: Search information: MySQL and Storage</td>
<td>Dates: 7</td>
</tr>
<tr>
<td>Internal task T3: Search information: JavaEE and Web App</td>
<td>None</td>
</tr>
<tr>
<td>Internal task T4: Search information: Cryptography and Signature</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project: Application for JPEG privacy implementation with XACML</th>
<th>WP ref: (WP#) 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major constituent: Desgin application</td>
<td>Sheet 2 of 5</td>
</tr>
<tr>
<td>Short description: Propose the design of application, estimate storage, databases, others…</td>
<td>Planned start date: 25-09-17</td>
</tr>
<tr>
<td></td>
<td>Planned end date: 15-10-17</td>
</tr>
<tr>
<td></td>
<td>Start event: 25-09-17</td>
</tr>
<tr>
<td></td>
<td>End event: 15-10-17</td>
</tr>
<tr>
<td>Internal task T5: Use JPEGMetadataEditor</td>
<td>Deliverables: None</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Internal task T6: Design structure Web App</td>
<td></td>
</tr>
<tr>
<td>Internal task T7: Create Database</td>
<td></td>
</tr>
<tr>
<td>Internal task T8: Definition of storage for images</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project: Application for JPEG privacy implementation with XACML</th>
<th>WP ref: (WP#) 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major constituent: Use of balana and privacy policies</td>
<td>Sheet 3 of 5</td>
</tr>
</tbody>
</table>
| Short description: Learn to use the Balana tool and design privacy policies | Planned start date: 15-10-17  
Planned end date: 10-11-17  
Start event:  
End event: |

<table>
<thead>
<tr>
<th>Internal task T9: Create policies for request.</th>
<th>Deliverables: None</th>
<th>Dates: None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal task T10: Do an a simple authorization</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project: Software prototype: Application for JPEG privacy implementation with XACML</th>
<th>WP ref: (WP#) 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major constituent: Programming the main application</td>
<td>Sheet 4 of 5</td>
</tr>
</tbody>
</table>
| Short description: Program the web application and use of the system. | Planned start date: 08-10-17  
Planned end date: 15-01-18  
Start event: 08-10-17  
End event: 10-01-18 |

<table>
<thead>
<tr>
<th>Internal task T11: First version web app</th>
<th>Deliverables: None</th>
<th>Dates: None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal task T12: Digital signature and encryption</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal task T13: Methodology for create policies from requierements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal task T14: Extend app for other use cases</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**1.4.2. Milestones**

<table>
<thead>
<tr>
<th>WP#</th>
<th>Task#</th>
<th>Short title</th>
<th>Milestone / deliverable</th>
<th>Date (week)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,2,3,4</td>
<td>Search information</td>
<td>-</td>
<td>07-10-17</td>
</tr>
<tr>
<td>2</td>
<td>5,6,7,8</td>
<td>Design</td>
<td>-</td>
<td>15-10-17</td>
</tr>
<tr>
<td>3</td>
<td>9,10</td>
<td>Balana and create policies</td>
<td>Policies examples</td>
<td>10-11-17</td>
</tr>
<tr>
<td>4</td>
<td>11,12,13,14</td>
<td>Program main app</td>
<td>Application</td>
<td>18-12-17</td>
</tr>
<tr>
<td>5</td>
<td>15,16</td>
<td>Test and documentation</td>
<td>Final report</td>
<td>25-01-18</td>
</tr>
</tbody>
</table>

**1.4.3. Gantt Diagram**

![Gantt Diagram](image-url)

*Figure 1.4.3: Gantt Diagram*
1.5. **Deviations and incidences**

No major incidences have been found in the project, but some work packages have had to be modified to spend more time learning the XACML standard, which has been one of the bases of the project.

Finally the initial idea has been complemented by improvements and other points of view generated during the project.

2. **State of the art**

In this project, it uses technologies and implementations such as XACML standard [1] for the generation of privacy policies, Balana [2] as the engine that implements the architecture described in XACML and is able to analyze the entire process, it adds encryption for metadata security, it also includes a digital signature to ensure authenticity and finally complemented with WSO2 IS [3], software used as an editor to create policies.

2.1. **XACML**

XACML (eXtensible Access Control Markup Language) was ratified as an OASIS standard in 2003 and the latest version 3.0 was ratified in January 2013. XACML is a standard that implements a language for access control policies based on XML, was designed to become a universal standard for describing who has access to which resources. It also defines architecture for the interpretation of the policies that are generated. It also covers the request-response control protocol, which is based on how to create requests and how to manage the responses.

As the graph shows, the architecture that defines [4] the standard has the following structure:

![Figure 2.1: XACML Architecture](image-url)
<table>
<thead>
<tr>
<th>PAP</th>
<th>Policy Administration Point - The system entity that creates or receive access policies</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDP</td>
<td>Policy Decision Point - Evaluate access requests against authorization policies before issuing access decisions</td>
</tr>
<tr>
<td>PEP</td>
<td>Policy Enforcement Point - Intercepts the user's request for access to a resource, demand a decision from the PDP to obtain the access decision</td>
</tr>
<tr>
<td>PIP</td>
<td>Policy Information Point - Entity that acts as a source of attribute values</td>
</tr>
<tr>
<td>PRP</td>
<td>Policy Retrieval Point - Stores XACML access authorization policies</td>
</tr>
</tbody>
</table>

2.2. **Balana**

Balana is an implementation of the architecture proposed by the XACML standard, therefore it defines tools to be able to work with the system described. The Balana project can be found in [2] where it consists of four modules:

1. Balana core – This is the actual implementation.
2. Balana samples – This contains the samples.
3. Balana Utils – This contains some utility methods in Balana.
4. Balana documentations – This contains docs.

2.3. **Encryption and Decryption**

The encryption of data and the use of encryption algorithms are fundamental for the security of an access control system, since they guarantee the invulnerability of communications between the devices that compose it.

Its use is essential to avoid attacks on the system, since they allow the information exchanged between the different elements to be completely undecipherable for non-users.

Advanced Encryption Standard (AES) [5] is one of the most widely used and secure encryption algorithms currently available. It started in 1997, when the NIST (National Institute of Standards and Technology) began looking for a successor to the standard DES encryption. The new AES encryption standard was officially announced in 2001.

The symmetric algorithm is based on several substitutions, permutations and linear transformations, each executed in blocks of data of 16 bytes. These operations are repeated several times, called "rounds". During each round, a unique circular key is calculated from the encryption key and incorporated into the calculations. The difference between AES-128, AES-192 and AES-256 is finally the length of the key: 128, 192 or 256 bits.

2.4. **XML Signature**

The need to guarantee the integrity, confidentiality and authenticity of the data has become an essential requirement. To solve these problems, standards such as XML Signature [6] have been developed.

XML Signature ensures the integrity of parts of transported XML documents.
It also provides authentication for data of any kind, whether it is in the XML that includes the signature or elsewhere. It can be applied to any digital content represented by XML. Primarily, XML Signature is to associate keys to the data.

XML Signature represents a system that through a digital signature allows offering authenticity of the data. The digital signature confirms the identity of the sender, the authenticity of the message and its integrity, without forgetting that the messages will not be disowned.

2.5. **WSO2 Identity Server**

WSO2 Identity Server is a product that allows us to manage the credentials and protocols of access to the resources and services of an entity through a single point of management. This solution allows us to manage all the internal authentication servers as if it were one only. WSO2-IS provides the most widely used access protocols in the market.

The product covers many tools for the management of privacy and security but in our case we are going to focus on the part related to XACML where it provides us with some policy editors and attributes management that we will use during the project.

### 3. Methodology / project development

The proposed solution for this project will be detailed, the objective is JPEG privacy implementation with XACML standard [7].

The project is divided into three phases in order to achieve the final objective, since apart from the privacy system that is the main purpose, we first need to design the privacy policies and then add them to the metadata.

- **Creating Policies**: Defines how privacy policies are created and designed.
- **Updating Metadata**: It is defined as adding the policies to the metadata of a JPEG image.
- **Privacy System**: The logic of the system that defines how a user requesting access to a JPEG image reaches its visualization through the integration of all components in an environment for user interaction.

#### 3.1. Creating Policies

The first phase of this project is the creation of policies, so we are going to dedicate this part to the treatment and logic of this process.
One of the most important points of this project is the creation of privacy policies based on the XACML standard. XACML defines three top-level policy elements: <Rule>, <Policy> and <PolicySet>

Policy sets contain different policies. Policies are based on rules, each rule can have the effect of permit or deny.

![Figure 3.2 Policies Structure]

An example in the figure above, the policy set includes four policies and each policy has two rules. XACML has a very high potential due to its easy scalability, it allows us to make very complex policies.

An important element is the combining algorithms that are used to solve conflicts between multiple policies and rules that are applied at the same time.

Combining algorithms must be used in:

- PolicySet elements
- Policy elements

The rule-combining algorithm defines a procedure for arriving at an authorization decision given the individual results of evaluation of a set of rules. Similarly, the policy-combining algorithm is used for a set of policies.

The most used Combining algorithms are listed below

- Deny-overrides (Ordered and Unordered),
- Permit-overrides (Ordered and Unordered),
- First-applicable
- Only-one-applicable.

Although there are other possibilities. (See XACML standard referenced for a full list of standard combining algorithms)

In the project, different policies were created without generating a set of policies, since it was not necessary for the final objective and we can work at the level of <Policy> as a
root element. So we are going to focus on the important element to make the decisions that are the rules.

A rule is the most elementary unit of policy and must be encapsulated in a policy. The main components of a rule are:

- **Description** – Documentation
- **Target** – Select applicable policies
- **Condition** – Boolean decision function
- **Effect** – Either “Permit” or “Deny”

**Description**: plain text with a brief description of the rule or policy that applies

**Target**: allows deciding if a rule applies to a request or not without evaluating the condition of the rule. In the case that it is NotApplicable, that rule or policy will have no effect for the response. If there are no other rules or policies that permit or deny, the response will be “NotApplicable”. It can be applied to different attributes. It is very important when creating Policysets and being able to filter which rules of all are evaluated according to the target.

**Condition**: this element sets the conditions that the request must satisfy to allow authorization. Attributes, functions and datatypes that are requirements of the designed policy are determined. It is possible to generate conditions also to deny access, so we have different ways of setting the rules.

**Effect**: The result of each rule is determined:

- If condition is true, return Effect value.
- If not, return Not Applicable.
- If error or missing data return Indeterminate.

### 3.1.1. XACML

In this chapter we are going to focus on the language with XACML code and to review how the project policies are created through code directly.

Once we have achieved a general knowledge of the XACML standard and the basic structure, we will see the creation of an XML policy using code to be used in our system.

In this case, the Eclipse editor has been used directly to create the policies. Eclipse is the software that we will also use to create the application to manage the privacy of the images.

The main structure of the policies designed is the following.

```
<Policy>
  <Description/>
</Description>
<Target/>
<Rule>
  <Condition/>
</Condition>
</Rule>
</Policy>
```
As we saw in the previous chapter, there is a <Policy> tag that encompasses the entire policy to be designed. The <Description> element will contain a brief explanation, and the <Target> that is used to define their scope, and finally the <Rule> with the element <Condition> inside, where we really determine what requirements the request has to satisfy to permit or deny access.

In order to complement these general tags and give functionality to our XACML structure code, we have syntax that we must use appropriately, there are many elements to use, but we will highlight those most used in our code and essential for designing policies.

Some of these elements as explained in the oasis XACML 3.0 standard.

“The <Apply> element denotes application of a function to its arguments.”
“<AttributeValue> contains a literal attribute value.”
“The <AttributeDesignator> element retrieves a bag of values for a named attribute from the request context.”

Within these elements, the DataTypes, AttributesId, Categories and FunctionsID are detailed. It allows us to complete the functionality of the defined rules.

**Example of policy with XML editor**

Below is one of the policies we have generated for this project in XML code. The rule is applied in a business case.

“Only workers of the trading department during the month of November can access the image.”

```xml
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
PolicyId="Policy1_TFG"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:permit-overrides"
Version="1.0">

<Description>All workers of trading department in November</Description>

Brief description about the use of this policy.

```
Next, the target of the policy is defined, it is required that the attribute department of the request be equal ("string-equal") to trading, in other case if the employee is from another department the policy will not be evaluated and it will return Not applicable for this request.

Also define the elements, MatchID, DataType, AttributeID, and Category with their respective URIS according to the requirement that must satisfy.

MustBePresent = "True" determines that it is an obligation to evaluate the target.

```
<Rule Effect="Permit" RuleId="PermitRule1">
  <Condition>
    <Apply FunctionId="urn:oasis:names: tc:xacml:1.0:function:and">  
        <Apply FunctionId="urn:oasis:names: tc:xacml:1.0:function:string-equal">
            <AttributeDesignator AttributeId="department">
                Category="urn:oasis:names: tc:xacml:1.0:subject-category:access-subject" 
                DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
        </Apply>
        <Apply FunctionId="urn:oasis:names: tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator AttributeId="MatchID">
                Category="urn:oasis:names: tc:xacml:1.0:subject-category:access-subject" 
                DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </AttributeDesignator>
            <AttributeDesignator AttributeId="DataType">
                Category="urn:oasis:names: tc:xacml:1.0:subject-category:access-subject" 
                DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </AttributeDesignator>
            <AttributeDesignator AttributeId="AttributeID">
                Category="urn:oasis:names: tc:xacml:1.0:subject-category:access-subject" 
                DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </AttributeDesignator>
            <AttributeDesignator AttributeId="Category">
                Category="urn:oasis:names: tc:xacml:1.0:subject-category:access-subject" 
                DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </AttributeDesignator>
        </Apply>
    </Apply>
  </Condition>
</Rule>
```

Now, the code focuses on the body of a particular rule, in this case the rule that will allow us to access the image if we are in the month of November and the employee is from the trading department.

In this particular case we have two conditions to combine, using the conditional AND indicated by "<Apply FunctionId = "urn:oasis:names: tc:xacml:1.0:function:and">. With this statement it is possible to combine two conditions in a rule.

Very similar to the target to establish that one of the conditions is the attributeID equal to department and trading as a value.
The other condition, in the month of November, but in this case dividing the condition in greater than the day 1-11-2017 and less than the day 30-11-2017. Applying the date-“greater-than” and “date-less-than” functions can be managed.

In this case, as a DataType, it is indicated as a date, with “DataType = http://www.w3.org/2001/XMLSchema#date”.

Lastly, an additional rule is added in this policy,

```
<Rule Effect=“Deny” RuleId=“DenyRule”/>
```

A rule is defined without conditions that will always return the Boolean value as true, and “Deny” effect.

This rule is usually added for the case where the other rules are false, the output has a Deny effect and does not appear as NotApplicable.

The “permit-overrides” algorithm will be imposed when a rule has a Boolean true and effect of rule permit, therefore it only has an effect in the case that the request does not satisfy any rule.

The definition of rules in XACML code can be complex for users without previous experience in this language, so a table has been proposed where the policy can be encapsulated in a visual and readable form in a simple way.

It can be seen that all the elements that are mentioned in the previous sections are indicated for design a policy, i.e: attributes, functions, datatypes, etc.

<table>
<thead>
<tr>
<th>Category</th>
<th>Atributte</th>
<th>Type</th>
<th>FunctionID</th>
<th>Value</th>
<th>Condition</th>
</tr>
</thead>
</table>

Policy defined with table:

<table>
<thead>
<tr>
<th>POLICY 1 - permit-overrides</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULE 1 - Permit</td>
</tr>
<tr>
<td>Subject</td>
</tr>
<tr>
<td>Environment</td>
</tr>
<tr>
<td>Environment</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RULE 2 - Deny</th>
</tr>
</thead>
<tbody>
<tr>
<td>No conditions</td>
</tr>
<tr>
<td>-</td>
</tr>
</tbody>
</table>

| - | - | - | - | - |

Table 3.2.1 : Defined Policy

3.1.2. WSO2 Identity Server

In this chapter we focus on the usefulness of the WSO2 Identity Server platform, which through its editors facilitates the creation of policies.

The generation of rules directly in XACML language can be complex for users who do not have knowledge about XML language. The syntax can be confusing to create simple policies and rules, so it was decided to use a tool to define policies in a simple way.
After researching about XACML policy editors, we found a software created by WSO2 called WSO2 Identity Server that allows to create these policies and contain some other functions to help the management of privacy in an organization.

To be able to use this software, you can download it from its website [10] and then follow the steps [11] for installation.

We execute the script to configure a server in local host and its tools for our own use, we can access the system through the URL: https://localhost:9443/carbon/

This software has different editors to generate rules, we see the options after navigating on Home -> Entitlement -> PAP -> Policy Administration -> Add New Policy

It has four editors that are valid for generating policies. In this project, the third option, the Standard Policy Editor, has been used.

The Standard Policy Editor is configurable and allows adding attributes, categories, datatypes, etc. It is flexible for our work if we want to have our own categories and attributes, and is also updated to the full version XACML 3.0.

In the image there is a first section to indicate the ID, the combining algorithm and the description of the new policy. After, three sections that are minimized, the first to put the
target, the second define the conditions and finally the possibility of adding obligations and advices.

We can see the example of the policy that we have made in XACML in the last chapter, but in this case generated through this Standard Policy Editor.

*<Policy> and <Description>*

![Figure 3.2.2.D: Description Policy Editor](image)

*<Target>*

![Figure 3.2.2.E: Target Policy Editor](image)

*<Conditions>*

![Figure 3.2.2.F: Conditions Policy Editor](image)

In the lower part of the section we can navigate through all the rules of the policy to configure them:
We see that this software allows us to create policies easily and without having knowledge about the XACML language. Once created, we can obtain it directly expressed in XML in the "edit in xml" section.

Finally we will show how we can edit the policy editor to add options such as categories, attributes, datatypes, etc.

For do this project, we have added to the category "Subject", other attributes such as "department", "year-entrance" or "system-attemps", the same in Resource with "max-attemps". It is shown in the previous figure.

To have these new attributes linked to the editor and generate correctly the URNs that we specify for generate a valid rules, we define them in the claims section of the software.
3.2. **Updating Metadata**

In this chapter we start the second phase of the project where the objective is to add the created policies to the JPEG metadata. We will also include a digital signature in them.

Another of the main points of the project is where we will store the privacy policies of each image. One of the most compact and logical solutions is to add them to the metadata of the image [12]. The standardization of the metadata segment is currently being discussed to include a bookmark exclusively reserved for the management of privacy and security of the JPEG image.

3.2.1. **JPSearch**

JPEG Search (or JPSearch) [13] is an International Standard developed by the JPEG Committee (ISO/IEC JTC1 SC29/WG1). It specifies mechanisms to search for images, including metadata description.

The purpose of the standard is to provide common information in metadata in order to different systems be able to interoperate. It provides a flexible architecture for the search metadata and an extension is being developed to include the management of privacy and security in APP3 segment in metadata.

Our purpose is to use an existing field in the metadata of the JPSearch standard called Rights Description that allows us to include the privacy policies in the image metadata.

### Figure 3.3.1 : JPSearch Segment

3.2.2. **JPEGMetadataEditor**

The JPEGMetadataEditor software is used to add the privacy policies in the JPSearch metadata. Initially developed [14] to modify metadata and later modified [15] for its adaptation to privacy metadata by means of the field Rights Descriptions.
The result provides a software with which it is possible to modify metadata and especially to take advantage of the Rights Description field to store a privacy policy with XACML code.

This program developed in java language had to be modified in this project to add the digital signature, since this feature was not included, in order to ensure the integrity and authentication of the metadata.

The following image shows the field where the XACML policy can be included with an example of privacy policy created for this project.

![Image of JPEGMetadataEditor](image)

Figure 3.3.2: JPEGMetadataEditor

3.2.2.1. Insert policy

To enter a policy is done using the textbox provided by JPEGMetadataEditor in the privacy section, as shown in the previous figure.

Everything added in this field is collected in the Rights Description field of the metadata. By attaching the XACML code of the policy in this space you can include the policy to the JPEG image.

This policy will be part of the metadata segment marker JPSeach called APP3, determined at the moment to contain the metadata privacy information.
**3.2.2.2. Insert XML signature**

Once the privacy policy has been introduced in this space, when updating the metadata with the "Update Metadata" button, an XML digital signature is added automatically in another field of the metadata.

The field in which we are going to include the signature is the labeling as "Keyword". It is defined in this project in this way, but it is currently being standardized which fields and which parts of the segment are going to really contain all this information about privacy in the future.

The digital signature XML is created in enveloped form. In this case, a signature is made using the DSA algorithm [16] and the hash function, SHA1, is added.

An example of a signature generated in the project is shown below.

```xml
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#"
  <SignedInfo>
    <CanonicalizationMethod
      <Algorithm>http://www.w3.org/TR/2001/REC-xmlenc10#WithComments" /></
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#ds-a11" />
    <Reference URI=""
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>QH1D/FyYb2YowoP7R3&F7AC9r4a==</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>MeVmpbqJn3WcWkrVo0sS5ha2TFV9DEeX069sAekr4xDrzwElGx1cv0IA==</SignatureValue>
  <KeyInfo>
    <SignatureValue>
      <KeyValue>
        <DSKeyValue>
          "P"/Ku26048yrom78s8QSShB4sF7ey7cEYI8646866R91195i759jQ7ex8u01mz3572zV99a7m8n1knFw==
        </P>
        "Q"/117dDaqc067G77h6zkn2TRUCM==
      </SignatureValue>
      <KeyInfo>
        <SignatureValue>
          "K"/Ku26048yrom78s8QSShB4sF7ey7cEYI8646866R911915i759jQ7ex8u01mz3572zV99a7m8n1knFw==
        </P>
        "Q"/117dDaqc067G77h6zkn2TRUCM==
      </SignatureValue>
    </SignatureValue>
  </KeyInfo>
</Signature>
```

**3.2.2.3. Encryption**

At this point it is detailed how the encryption of the system is generated for greater security. The two fields of the metadata that we are going to encrypt are those that contain the privacy policy <RightsDescription> and the one that contains the digital signature <Keyword>.

Both fields are encrypted with symmetric encryption where the AES key is generated randomly coded with Base64 and the respective field is encrypted for each part, adding SHA 256 hash algorithm and finally encrypted with the AES standard.

The following image details how both fields are encrypted in the same way, we also have to remember as indicated in the previous chapter that the signature has its own asymmetric encryption through DSA + SHA1. It is done internally in the field <SignatureValue>
After the first two phases of the project, in this last phase, the application called Rights Validator will be designed, which will serve to close the project cycle and have a system capable to work with the designed policies and that we obtain through the metadata and implementing the Balana system for its validation.

To start developing this last part of the project we take into account an overview system as shown in the following figure.

The figure shows a horizontal central axis where the most basic structure of the system is defined; a user accesses the application and obtains an image.
For this process, the top part includes two main resources, storage for the images and databases. In addition, the core of the system is the Balana authorizer which will permit or deny access to the images requested by the user.

3.3.1. File Storage
A space is defined to store the images, specifically on the local disk. There are other solutions such as an external server or directly in the cloud to give the system more flexibility.

3.3.2. Databases MySQL
The necessary databases are also defined to store the information of both, the user who accesses the system and the information of the images. In this project we have worked directly with "MySQL Workbench" [8], a desktop application to manage MySQL [9] databases.

3.4. Core Structure Rights Validator App
The following schema shows the logic that has the core of the “Rights Validator” application developed in this project, starts with the selection and extraction of the metadata of the image to obtain the policy until the authorization through the Balana implementation to get the visualization of the desired image.

3.4.1. Login and Image Select
As an input to our system, a screen to perform the login through a username and password is displayed.
First checking that the user trying to access the system is in the user database. If the access process is correct, the images contained in the database are shown to the user and so he can select the desired image to try to visualize it.

3.4.2. Extract Metadata

The user selects the desired image by clicking on the icon and the logic is executed to check if the user is valid to visualize the image.

The first step is the extraction of metadata from the image, which is provided by the JPSearch standard through the defined segment.

Once the bytes of the metadata are decoded to plain text, an XML format of all the metadata of the image is obtained, including the `<RightsDescription>` and `<Keyword>` fields.

We see an example of extracted metadata from image in appendix A

3.4.3. Obtain Policy and Signature

Once we have extracted the metadata from the image, from the metadata we must obtain the required fields.

Previously commented that the `<RightsDescription>` fields that contain the privacy policy and `<Keyword>` that contains the digital signature are required, so by parsing the metadata in XML format, the two pieces of information are obtained in an encrypted form.

```
<RightsDescription>
  <RightsDescriptionInformation>
    <ActualRightsDescriptionReference/>
    <ActualRightsDescriptionReference/>
  </ActualRightsDescriptionInformation>
  ...
  ...
  b6mLlYHnN5Mc+cvLJnH5Mc+u49+f2U0/hPMGyc2duQgg6m6YmjhWj0X91VyiA6f1kl
  M1b/CCNfLikj2MHDvCANN1k1mNR+nH5Mc+HjU36dDfE+VWFscT+Paj3LRCYp/xILwqQ==
  </ActualRightsDescription>
</RightsDescription>
```
3.4.4. Decryption

The next step is the decryption of both the policy and the signature, using the inverse process to the one that performs the encryption in the JPEGMetadataEditor software. Therefore, the privacy policy and the digital signature are available in XML format as originally created. The application can now send to the validator the signature and the privacy policy to Balana for the validation and authorization process.

3.4.5. Validate XML Digital Signature

The digital signature is sent to the xml signature validator, where with the appropriate information contained in the fields <SignedInfo> and <KeyValue> it is possible to decipher the key for verification. In case the signature is valid, it will save a variable with a boolean to complement the final decision of the system. The XML digital signatures are important because add authentication, non-repudiation and data integrity, although the access permission is done through the privacy policy.

3.4.6. Generate Request

Another of the most important points of the system is the creation of the request that is compared with the privacy policy to permit or deny the visualization of the image selected. The request is generated in a very similar way to the privacy policies with elements defined in the XACML standard. The main element is <Request>, within this element you can define the attributes that are required when defining a request according to the system and the designed policies.

For the generation of the request, this system is fed from the database, where three attribute classes are obtained.

- Attributes of the user (i.e : Name, Age, Department ...)
- Attributes of the image (i.e: Name, number of accesses ...)
- Environment attributes. (i.e : Date, time ...)

This can be related to some of the attributes that are specified in the XACML standard: Subject (User), Resource (Image), Environment (Environment), Action (view)

User attributes are saved when logging into the system. On the other hand, the attributes of the image are generated when selecting the desired image.

Finally, the attributes related to the environment are generated from the system itself. It can be summarized that the system has automatically made the request with the user's access information, the selected image information and system data information.
3.4.7. Balana

Balana is an open source XACML implementation. This means that it is designed to fully support the standard, defining its data-flow model as specified.

The way that is used in this project is to indicate to Balana the directory where we have saved the policy extracted from the image and pass the request generated for the evaluation. In summary, the privacy policy extracted from the image and the request is delivered to Balana.

Balana establishes the implementation with the architecture defined by point 3.1 of the XACML standard:

![Figure 3.4.7: XACML standard architecture](image)

Balana through its architecture based on the XACML standard, is able to verify that the request is valid or not for the privacy policy saved and will return a response with the final decision.

An example of response from Balana when the result is “Permit”:

```xml
<Response xmlns="urn:oasis:names:tc:xacml:3.0:core:schema-md-17">
  <Result>
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
```
3.4.8. Display and Updates

In the final part of the application, a filtering is performed with the results of the digital signature and Balana's response about the privacy policy.

If the digital signature fails, an error appears with the text "Signature error" and the image is not displayed. On the part of Balana there is a decision in his response. Deny, Not Applicable or Indeterminate causes an error message where it shows the reason:

- Deny: the decision is in normal conditions, denying access because it does not comply with the privacy policy attributes.
- Not Applicable: The target of the policy does not match the request element, so the privacy policy does not apply to it.
- Indeterminate: Error in the syntax of the request or policy. It is possible a bad structure of elements or erroneous attribute.

![Error](image)

**Figure 3.4.8.A: Deny display**

Only in the case that the digital signature is valid and Balana's response is "Permit" the user will be shown a message indicating that have authorization to see the image.

![Correct access](image)

**Figure 3.4.8.B: Permit display**

Finally, some parameters of the database that are dynamic are updated as an example the times an image has been shown. So we can create policies that contain variable and updatable restrictions.
4. **Results**

The process of this project provides the "Rights Validator" application capable of acting as a privacy system for JPEG images through the policies defined in the XACML standard.

A system capable of validating and authorizing a request on an image of the database is implemented correctly, and allows viewing only if the conditions stipulated in the privacy policy contained in its metadata are correctly fulfilled.

For this purpose, another application defined in previous projects "JPEGMetadataEditor" is used to proceed with the integration of the privacy policy into the metadata. The digital signature has been adapted to this existing project since it did not include it.

In order to verify the satisfactory result of the application, two main points of the project have been analyzed: Design privacy policies and evaluate them in the authorizer.

4.1. **Privacy policy design**

It has been possible to design any type of privacy policy that has been needed through the XACML standard language, using a wide variety of elements provided by the standard. Versatility of this language has been proven by giving different policies with all kinds of attributes, datatypes, functions and decision algorithms. On the other hand, XACML native code is quite complex for a user that does not know the XACML language, another alternative has been presented that is to use an editor to create privacy policies. It has been used and demonstrated that can be created through open source software (WSO2 Identity Server) that allows us to create policies in a more friendly way.

For this reason, we believe that the expected results for the creation of policies have been fulfilled since we have not had any problems defining any type of requirement to formulate a policy. We have also proposed different ways to create policies.

In appendix B we find some of the main policies defined in the project.

4.2. **Rights Validator App**

The final result will determine if the application is able to correctly associate the available resources: images, databases, Balana implementation and the final decision for display or not the image.

To determine a result of the sequence that the application follows, it has been decided in each step that it takes to show the result it obtains in order to go through the entire architecture of the application and check the final decision according to our expectation.

Since the possible combinations for the verification are infinite, it has been decided to establish some concrete examples of inputs and to observe if the output is the expected one.

In all cases generated the expected response has been obtained, cases of "Permit" and "Deny" have been tested. You can check these tests performed in the next section.

In the appendix C you can check some examples of execution of the application.
4.3. **Test 1**

To perform a test in order to obtain valid and verified system results, a logic of access to the system and its respective requests to the images is defined.

In our environment created for the use case of a company and its employees there are 4 images defined with 4 different privacy policies. The first thing will be to know these policies and what users we have in the database.

The 4 images defined, with the purpose of their privacy policies are:

- **Company1.jpg**: All users of the trading department can access in the month of January 2018.

- **Company2.jpg**: Employees with year of entry to the company equal to or less than 2013 can access to the image between 9:00 a.m. and 5:00 p.m.

- **Company3.jpg**: It will be shown only to the first 3 users who access as trading department and senior role.

- **Company4.jpg**: Users with less than 26 years old or those with less than 2 accesses to the system.

On the other hand, the two tables that we have in the database for the company system are a reference for users and another for images.

<table>
<thead>
<tr>
<th>idusers</th>
<th>user</th>
<th>pass</th>
<th>department</th>
<th>role</th>
<th>age</th>
<th>year_entrance</th>
<th>system_attempts</th>
<th>company</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>miguel</td>
<td>123</td>
<td>trading</td>
<td>senior</td>
<td>28</td>
<td>2009</td>
<td>0</td>
<td>xacnl</td>
</tr>
<tr>
<td>2</td>
<td>laime</td>
<td>123</td>
<td>finance</td>
<td>junior</td>
<td>36</td>
<td>2015</td>
<td>0</td>
<td>xacnl</td>
</tr>
<tr>
<td>3</td>
<td>lara</td>
<td>123</td>
<td>trading</td>
<td>junior</td>
<td>26</td>
<td>2014</td>
<td>0</td>
<td>xacnl</td>
</tr>
<tr>
<td>4</td>
<td>carlos</td>
<td>123</td>
<td>marketing</td>
<td>junior</td>
<td>25</td>
<td>2015</td>
<td>0</td>
<td>ext</td>
</tr>
</tbody>
</table>

4.3.1 **Users company database**

<table>
<thead>
<tr>
<th>idimages</th>
<th>name</th>
<th>max_attempts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Companyv1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Companyv2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Companyv3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Companyv4</td>
<td>0</td>
</tr>
</tbody>
</table>

4.3.2 **Images company database**

With all this information we define a sequence of logins and accesses to images, we will note the result we expect in each of the iterations, finally we will complete the table by doing the sequence in our system and compare the expected result with the obtained one that should be the same.

The sequence to follow will be:

- Login user miguel -> company1.jpg -> company2.jpg -> company3.jpg -> company4.jpg
- Login user laura -> company4.jpg -> company1.jpg -> company3.jpg
- Login user carlos -> company3.jpg -> company4.jpg -> company1.jpg
- Login user carlos -> company4.jpg
To assess the results we have defined a table with different fields to consider. The first is the type of access, if it is a login or an image request. Second the name of the resource or user who accesses. The next two show the expected and the obtained result. We also add the variables that will be updated in the system dynamically. We have the variable number of accesses to the system for the user and the maximum visualizations for an image, thus we create a more complex and sophisticated system. Finally the notes field, where the reason for the failed access is noted.

The table is as follows:

<table>
<thead>
<tr>
<th>Type</th>
<th>name</th>
<th>Result expected</th>
<th>Result obtained</th>
<th>Tables updates</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login</td>
<td>miguel</td>
<td>-</td>
<td>miguel_sysatt = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Company1</td>
<td>Permit</td>
<td>Permit</td>
<td>company1_maxatt = 1</td>
<td>Time&gt;17:00h</td>
</tr>
<tr>
<td>Request</td>
<td>Company2</td>
<td>Deny</td>
<td>Deny</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Company3</td>
<td>Permit</td>
<td>Permit</td>
<td>company3_maxatt = 1</td>
<td>Rol = junior</td>
</tr>
<tr>
<td>Request</td>
<td>Company4</td>
<td>Permit</td>
<td>Permit</td>
<td>company4_maxatt = 2</td>
<td>Rol = junior</td>
</tr>
<tr>
<td>Login</td>
<td>laura</td>
<td>-</td>
<td>laura_sysatt = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Company4</td>
<td>Permit</td>
<td>Permit</td>
<td>compant1_maxatt=2</td>
<td>Departament = marketing</td>
</tr>
<tr>
<td>Request</td>
<td>Company3</td>
<td>Deny</td>
<td>Deny</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Company1</td>
<td>Permit</td>
<td>Permit</td>
<td>company4_maxatt = 3</td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Company4</td>
<td>Deny</td>
<td>Deny</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Login</td>
<td>carlos</td>
<td>-</td>
<td>carlos_sysatt = 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Company3</td>
<td>Deny</td>
<td>Deny</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Company4</td>
<td>Deny</td>
<td>Deny</td>
<td></td>
<td>system_att &gt;1 and age &gt;26</td>
</tr>
<tr>
<td>Login</td>
<td>carlos</td>
<td>-</td>
<td>carlos_sysatt = 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Company4</td>
<td>Permit</td>
<td>Permit</td>
<td>company4_maxatt = 4</td>
<td>under 26 OK</td>
</tr>
</tbody>
</table>

**4.3.3 Test summary table**

After the simulation in our system gives us a result of 100% success as we expected. This test varies according to the time and day that the sequence is performed since we also have conditions such as date and time. In this case it was done on 01/20/2018 at 20:17

Tables with the variables updated at the end of the test:

### Users updated company database

<table>
<thead>
<tr>
<th>users</th>
<th>user</th>
<th>pass</th>
<th>department</th>
<th>role</th>
<th>age</th>
<th>year_entrance</th>
<th>system_attempts</th>
<th>company</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>miguel</td>
<td>123</td>
<td>tradnc</td>
<td>senior</td>
<td>28</td>
<td>2009</td>
<td>2</td>
<td>xacrl</td>
</tr>
<tr>
<td>2</td>
<td>laime</td>
<td>123</td>
<td>finance</td>
<td>junior</td>
<td>35</td>
<td>2015</td>
<td>0</td>
<td>xacrl</td>
</tr>
<tr>
<td>3</td>
<td>laura</td>
<td>123</td>
<td>tradnc</td>
<td>junior</td>
<td>25</td>
<td>2014</td>
<td>1</td>
<td>xacrl</td>
</tr>
<tr>
<td>4</td>
<td>carlos</td>
<td>123</td>
<td>marketing</td>
<td>junior</td>
<td>25</td>
<td>2015</td>
<td>2</td>
<td>ext</td>
</tr>
</tbody>
</table>

### Images updated company database

<table>
<thead>
<tr>
<th>images</th>
<th>name</th>
<th>max_attempts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Comano1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Comano2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Comano3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Comano4</td>
<td>4</td>
</tr>
</tbody>
</table>
4.4. **Another environment defined in the system**

In the next section of the results, we want to show that the system designed for the JPEG image privacy management is universal and can be adapted to any type of environment that your service requires. Therefore, for any situation the general case is to have images and want to protect them through these privacy policies designed in XACML.

Then another case is established with different conditions and resources, since we are going to use other policies as well as other images and we will also obtain information from other databases to complete the requests to the system.

The previous environment was generated as a company with its employees and their own images that were uploaded to an internal intranet to share them.

In this case we are going to suppose that some photographers want to sell their images to users through the payment of a subscription and we want to control it with a system of privacy policies based on XACML.

![Diagram](image)

**4.4.1 Schema of environments**

In the graph we see how to access another generated system, with the same logic as the previous one, but changing the input users, the databases and the images hosted with different privacy policies.

Some parts have had to be duplicated to work independently and thus obtain greater abstraction from the rest of the system. The classes that create the request and access the data of the databases are different, but the main core of the system is the same and has not been modified.
Now in the login you can choose the environment in which you want to work.

4.4.2 New login page

Selecting the "Rights Validator Photography" option will take us to the new environment created for your policy evaluation.

The images available in this environment are also displayed.

4.4.3 Images of new environment

As in the previous case, we will test the new environment in a sequence for the test.

4.5. Test 2

In this case we are going to simulate a sequence defined in the same way as in the first test carried out in the project but with the new users, images and information.

We define the three privacy policies designed in this environment:

- Photo1.jpg: Access for users from France or Spain if they request access in the last 15 days of January
- Photo2.jpg: Users who have paid basic subscription can access if they are from the US and users who have paid the premium subscription from anywhere.
- Photo3.jpg: Any user who has paid their fee can access.
The following sequence has been generated to validate the results:

Login user miguelimg -> Photo1.jpg -> Photo2.jpg -> Photo3.jpg
Login user jaimeimg -> Photo1.jpg -> Photo2.jpg -> Photo3.jpg
Login user lauraimg -> Photo1.jpg -> Photo2.jpg -> Photo3.jpg
Login user carlosimg -> Photo1.jpg -> Photo2.jpg -> Photo3.jpg

In this case in the table results there is no field of updatable tables, because in this case we do not have variables that are updated by access or visualization, we certainly have variables that depend on a payment system to update, which would have to be done through another external service to confirm. In this case we have simulated it manually.

<table>
<thead>
<tr>
<th>Type</th>
<th>name</th>
<th>Result expected</th>
<th>Result obtained</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login</td>
<td>miguelimg</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Photo1</td>
<td>Permit</td>
<td>Permit</td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Photo2</td>
<td>Permit</td>
<td>Permit</td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Photo3</td>
<td>Permit</td>
<td>Permit</td>
<td></td>
</tr>
<tr>
<td>Login</td>
<td>jaimeimg</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Photo1</td>
<td>Permit</td>
<td>Permit</td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Photo2</td>
<td>Deny</td>
<td>Deny</td>
<td>No paid free</td>
</tr>
<tr>
<td>Request</td>
<td>Photo3</td>
<td>Deny</td>
<td>Deny</td>
<td>No paid</td>
</tr>
<tr>
<td>Login</td>
<td>lauraimg</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Photo1</td>
<td>Deny</td>
<td>Deny</td>
<td>EEUU no europe</td>
</tr>
<tr>
<td>Request</td>
<td>Photo2</td>
<td>Permit</td>
<td>Permit</td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Photo3</td>
<td>Permit</td>
<td>Permit</td>
<td></td>
</tr>
<tr>
<td>Login</td>
<td>carlosimg</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Photo1</td>
<td>Permit</td>
<td>Permit</td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>Photo2</td>
<td>Deny</td>
<td>Deny</td>
<td>No paid premium</td>
</tr>
<tr>
<td>Request</td>
<td>Photo3</td>
<td>Deny</td>
<td>Deny</td>
<td>No paid</td>
</tr>
</tbody>
</table>

4.5.3 Test Summary Photography Table

It has achieved again 100% of expected results comparing with the obtained results, this test has been carried out in the following conditions: on 01/21/2018 at 00:27

As it has been highlighted in all the sections of the results, the overall and functional results of the system are satisfactory.
5. **Budget**

For this project, 542 hours have been worked. The software used is open source and free so no additional costs are added on this concept.

The salary of a junior engineer is in the current market around € 11 per hour, so this amount has been established in the table below.

<table>
<thead>
<tr>
<th>Position</th>
<th>Amount</th>
<th>Hours</th>
<th>Price per hour</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junior engineer</td>
<td>1</td>
<td>542</td>
<td>11 €</td>
<td>5.962 €</td>
</tr>
</tbody>
</table>

Table 5: Budget table

The personal computer is used so there are no hardware costs for estimating the budget.

If you want to develop a global system with access from any terminal you would have to pay for a server and possibly some extra service such as a database or file storage in the cloud. The costs have been ignored since it has been worked as localhost.

6. **Conclusions and future development**

In the first part, the project started with the analysis of privacy policies from studying the XACML language, different policies and different forms of creation have been developed.

As a second part, the methodology for adding the policies and the digital signature to the metadata is generated from an existing application developed in previous projects. It has helped us to build the base of our “Rights Validator” application, since we needed JPEG images with privacy policies integrated in their metadata.

Finally, in the last part the application that defines the title of this project has been generated, an application for the privacy of JPEG images implemented from XACML, where we can validate the policies generated for different scenarios.

In conclusion, the main objective has been achieved, the application that allows us to manage the privacy of the images. It has also achieved the main goals that were originally proposed and others that were subsequently added, as a summary of these; a complete functional system, use XACML language, evaluate different incoming requests, integrate policies in the metadata, make practical examples of system use and include some security mechanisms such as encryption and digital signature.

For the future work, a more full system can be proposed that integrates the three parts treated, the policy creation, add these in the metadata and the validation system. The three modules in a single interface to be able to perform all the parts required by the user who accesses.

For this work you could establish an effective way to create a policy so that a user can easily incorporate it into their source image. Several tables/graphics are proposed during this project. These can be manipulated in a friendly way by a user who does not know the XACML language for the configuration of the policy in a visual form adding the fields
directly. Then it would have to convert to XACML code automatically from the table in order to integrate correctly in Balana.

Also take into account the source image so that the generated policy is automatically added to the metadata in the segment designed for this purpose of privacy and security. The signature should be added in the same way.

Table that collects attributes, functions, datatypes and everything necessary for the configuration of a policy in a friendly way.

<table>
<thead>
<tr>
<th>POLICY 1 - permit-overrides</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RULE 1</strong> - Permit</td>
</tr>
<tr>
<td><strong>Subject</strong></td>
</tr>
<tr>
<td>Environment</td>
</tr>
<tr>
<td>Environment</td>
</tr>
<tr>
<td><strong>RULE 2</strong> - Permit</td>
</tr>
<tr>
<td><strong>Subject</strong></td>
</tr>
<tr>
<td>Environment</td>
</tr>
<tr>
<td><strong>RULE 3</strong> - Deny</td>
</tr>
</tbody>
</table>

Table 6.A: Policy Configuration Table A

Another type of distribution for the table that the user could modify easily.

Table 6.B: Policy Configuration Table B
Bibliography


Appendix A

METADATA:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ImageDescription xmlns="JPSearch:schema:coremetadata">
  <Identifier>P5</Identifier>
  <Modifiers>
    <GivenName>Miguel</GivenName>
    <FamilyName>Castillo</FamilyName>
  </Modifiers>
  <Creators>
    <GivenName>Laura</GivenName>
    <FamilyName>Vilches</FamilyName>
  </Creators>
  <Publisher>
    <PersonName>
      <GivenName>Miguel</GivenName>
      <FamilyName>Castillo</FamilyName>
    </PersonName>
    <OrganizationInformation>
      <Name>TFG</Name>
      <Address>
        <Name>MCN</Name>
        <Description>MCN Company</Description>
      </Address>
    </OrganizationInformation>
  </Publisher>
  <CreationDate>2017-11-24T18:59:14.819+01:00</CreationDate>
  <ModifiedDate>2017-11-24T18:59:14.834+01:00</ModifiedDate>
  <Description>Beach</Description>
  <RightsDescriptionInformation>
    <Description>XACML3</Description>
    <ActualRightsDescriptionReference/>
  </RightsDescriptionInformation>
</ImageDescription>
Menorca

Prueba

0.0 0.0 0.0

2200 713 3200 1463

Its Miguel
Appendix B

Policy Image 1

```xml
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
PolicyId="Policy1_TFG"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:permit-overrides"
Version="1.0">
  <Description>All workers of trading department and month</Description>
  <Target>
    <AnyOf>
      <AllOf>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            trading
          </AttributeValue>
          <AttributeDesignator AttributeId="department" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
        </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <Rule Effect="Deny" RuleId="DenyRule"/>
  <Rule Effect="Permit" RuleId="PermitRule1">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeDesignator AttributeId="department" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            trading
          </AttributeValue>
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
          <AttributeDesignator AttributeId="department" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-greater-than">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
            </Apply>
          </Apply>
        </Apply>
        <Apply FunctionId="http://www.w3.org/2001/XMLSchema#date">
          2017-11-01
        </Apply>
      </Apply>
    </Condition>
  </Rule>
</Policy>
```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
PolicyId="Policy2_TFG"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:deny-overrides" Version="1.0">
  <Rule Effect="Permit" RuleId="PermitRule1">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-greater-than">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
              <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:environment-id"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
            </Apply>
          </Apply>
        </Apply>
      </Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-less-than">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
          <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:environment-id"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment"
DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
        </Apply>
      </Apply>
    </Condition>
  </Rule>
</Policy>

Policy Image 2

<!--Policy Image 2-->

<Policy xmlns="urn:oasis:names:tc:xacml:1.0:function:and">
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-less-than">
      <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-date"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment"
DataType="http://www.w3.org/2001/XMLSchema#date" MustBePresent="true"/>
    </Apply>
  </Apply>
</Policy>
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
PolicyId="Policy3_TFG"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:permit-overrides"
Version="1.0">
  <Target/>
  <Rule Effect="Permit" RuleId="rule1">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-less-than">
            <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
DataType="http://www.w3.org/2001/XMLSchema#string"
MustBePresent="true"/>
          </Apply>
          <AttributeValue>3</AttributeValue>
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
          <AttributeDesignator AttributeId="year-entrance"
Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
DataType="http://www.w3.org/2001/XMLSchema#string"
MustBePresent="true"/>
        </Apply>
        <AttributeValue>2013</AttributeValue>
      </Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <AttributeDesignator AttributeId="year-entrance"
Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
DataType="http://www.w3.org/2001/XMLSchema#string"
MustBePresent="true"/>
        <AttributeValue>2013</AttributeValue>
      </Apply>
    </Condition>
    <Rule Effect="Deny" RuleId="DenyRule"/>
  </Rule>
</Policy>
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
  <AttributeDesignator AttributeId="http://wso2.org/claims/role" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"></AttributeDesignator>
  <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">senior</AttributeValue>
</Apply>

<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
  <AttributeDesignator AttributeId="department" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"></AttributeDesignator>
  <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">trading</AttributeValue>
</Apply>

</Condition>
</Rule>

<Rule Effect="Deny" RuleId="DenyRule"></Rule>

</Policy>

Policy Image 4

<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyId="Policy4_TFG"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:permit-overrides" Version="1.0">
  <Target/></Target>
  <Rule Effect="Permit" RuleId="rule1">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-one-and-only">
            <AttributeDesignator AttributeId="http://wso2.org/claims/age"
                                    Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                    DataType="http://www.w3.org/2001/XMLSchema#integer"
                                    MustBePresent="true"></AttributeDesignator>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">26</AttributeValue>
          </Apply>
        </Apply>
      </Apply>
    </Condition>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-one-and-only">
        <AttributeDesignator AttributeId="http://wso2.org/claims/age"
                                Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                DataType="http://www.w3.org/2001/XMLSchema#integer"
                                MustBePresent="true"></AttributeDesignator>
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">26</AttributeValue>
      </Apply>
    </Apply>
  </Rule>
</Policy>
<AttributeDesignator AttributeId="system-attempts"
Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
DataType="http://www.w3.org/2001/XMLSchema#integer"
MustBePresent="true"></AttributeDesignator>
</Apply>
<Value>
<AttributeValue>
<Apply>
<AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#integer">2</AttributeValue>
</Apply>
</Value>
</Apply>
</Condition>
</Rule>
</Policy>
Appendix C

C.1 Example A

STEP 1 Metadata Extraction

META DATA:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ImageDescription xmlns="JPSearch:schema:coremetadata">
  <CreationDate>2017-11-19T21:31:20.827+01:00</CreationDate>
  <ModifiedDate>2017-11-19T21:31:20.829+01:00</ModifiedDate>
  <Description></Description>
  <RightsDescription>
    <RightsDescriptionInformation>
    </RightsDescriptionInformation>
    <Description>XACML3</Description>
    <ActualRightsDescriptionReference></ActualRightsDescriptionReference>
  </RightsDescription>
</ImageDescription>
```

`/f9xg8hiG6fCqjOHOPkW`
STEP 2  Encrypted RightsDescription:

Policy encrypted:

<xKeyword>
O5dxywYWkrvyt09xbD5a1v11WOMb0yj1Vs78Xeh9cYjX5rCd5sUa4HhMtq+NI
JSOE+e0B1apeskq8f8pl5coGc4A5u6asSdycAxo0f1L92+6cEsU1Ey1ls1ALUIMdBeWY
rupXQ/2bp34dWQ09ppBo9eI1JF0mFg9p9x0z/ZKUWEI1NL23jAxPjt+oXpjkp9pxZRoq1I/
0koBoxhGR7xv5SQKUXeI77539dxQFcmUb9Kc3w35fBpct1l7pEwnJZJI3bvc5vdjWxAYG
BrtSmrJHDGGWaqutUbNgnp0UF37nFupLQrzRszRJ/Ku8Q6cFdg4/4AAlnp2z5Rdn6+YAMBl
DcGlqzPjWb3e3/g8SYPq5/pLCGb/pgqPj+0FVWYMs6mLPCmPaiUalubuBYXCGMkboXLBn
OvhFqVqIPOLEJERE0sTNDjvhsEBVnPrK/psTylKeIaZrnVQNob1+nuyThbxd36/8qAp9Tsdm
91tQRFCthM1mcImVxU3BflC1XKZ6m5o9h1r/bge+bkvXNA4CtWclcjtT1RloAYEJVR5jA6M4
Djk7kRqDgC5l1YiS56ezFwHyxaXiCeEDc/bmaUYE7EsEso0F/BoYyi6456z/net/4de0yoyNF
hMJdhp/xZ72YMc4zwPA2z8yTlCoGo1/jgtQ8+iDTha5WUsm13h09OWfWy3zhPgrfKk
Z8ZugQ/EfuSMnpCnx51L7bUjji6skECF3CH3GAYFTvjsWAAKu5d1fBIJLcb1b5P934Wo
cc7eza5y5U5wrLWf5oVtCoxHlN0W/952KraQoNoEmxV++av1NpGFd+jyA1kPR
CkybFwPmXGwDxworFMPvQr7vSwQRI7465kCwunYJXZXLqczTs1cjKJuR03YgMmayXkq4Xzg
RnpnqvlqVlhbXPE8px10zFXFpVsv9wUMg4+g254XO/1SQWRMzTTAt741nPjQXqFbPv1lyRUpbInC
C+iE4VjlhbjVn299sDixIUm9q9IYq/vq5o/Tx66y0vEm7Qvyg/7jCjC30Fouc
AjQjsoCihihvYMGaAq0vOVMAXlsrXqvoS5ILL05bQn/T0VizuAvbdP33bEfbBp3g86OX6FB8z1
IHg0OBQbonglnqSYK91M28l5WKNY77xbKppkww2/R81k4Q0spjBq735SRUjuw3IqVDr8I
TMWmArZOSC/7kiqj7r14y9pQa1BuD0Zxsz0yNgncFTJ53mluxzzi1+nBYaAM0WT4Wk2C5t
Xbf2466KmjaOu0foLGe4wllm+48QyGr511m6L4kFMAhag/bhqRCDaRqYb2Y18CBTrt8Ck
RAAKa1gN2EBeo+X9zEd3+820t3XCCBytzJv+sJrClpW0240xKKBvRFAHw9pRE8UXeys/cA=
</Keyword>

<Title></Title>

<LabelDefinition></LabelDefinition>

<LabelValue></LabelValue>

</RightsDescription>

<OriginalImageIdentifier>
<OriginationOfID></OriginationOfID>
</OriginalImageIdentifier>

<GPSPositioning
longitude=0.0
latitude=0.0
altitude=0.0/>

<Height></Height>

<Width></Width>
STEP 3 Decrypt <RightsDescription>:

Clean Policy:

<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17">
  <Target/>
  <Rule Effect="Permit">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-less-than">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
          <AttributeDesignator Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
        </Apply>
      </Apply>
    </Condition>
  </Rule>
</Policy>
<AttributeDesignator AttributeId="http://wso2.org/claims/role" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"></AttributeDesignator>
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">senior</AttributeValue>
</Apply>
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
<AttributeDesignator AttributeId="department" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"></AttributeDesignator>
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">trading</AttributeValue>
</Apply>
</Condition></Rule><Rule Effect="Deny" RuleId="DenyRule"></Rule>
</Policy>

STEP 4 Encrypted <Keyword>:

Signature encrypted:
05dxyywVkryxt09xbD5AVlITWOMlbv0j1Vs78Xehv9cYjXSrdCnsU4aHhmTq+N1iJSOE+e0B1aepskg8f8p15c0gOch4A5uY6asSdyasAxf01r9z+6cEsU1EYi1sAIULMIdBewYrupXQ/2bp3dWqQ9vPbg9ieJFAOFmg9P9x0x/ZKWUEiNL23jAxPjt+jvXLPjk9pxZRoqI/0koBobbGR TxvD5GUxuSL7T539DqGcFMbbyKc3w35FzbPctIPe7EPnJZ Xi3b3u3VcdjJWXaYgPbtsMjHDDGWawqruNhNgp0UF53TnFUIgRzszRG/KuQ6CqFdg4+4TAinpp2sRdn6+YaMBIDcGIGZpJWv3e3/q8SYgq5/plcGB/fqPij+0FWYMsS67LyRoPAlAuLubuBYXGMQkxOlxhNohvFPVIFOEJERE0sTNDjvhsEBVghKP+wTlykElAzrVnQANob1+nuYTH4r3D3/8qAp9TSdm9iQRFcThM1McImVxU3BfMC1XZK6xM091lr/b+ekvXNA41CTaWlcCjiTlRoAYEJVo5JAM4DjkTrQDgc5ClivISS6fZWyXaXOicEdc/hMaYJEsEs0/OF/BoYyi6456z/net/4/deOy0yNfHMjdhnp/x2PFycM4zwrPB2s1jyT1CoGole/gjPtq8+iThe5UNwscm3H09wOFwY3zPhgrRFkkZZugG/EfusMnpCXX13L78bUuji+6KCF3H33GATFTVjsWaAku5dlFBiJWClb5Pa934Woce7zvau7U5pWRlvtWwvF5o8UGCoxInL0Nw/9S2KRAQmOQmOExVx++av1NpftGD+jrAy1kPCRkybqFvMzXGWDaorMFEqv7/7qSx1R1746bkTZwnyJZD1QLQmTz1CjCHKRU03YGmANXg44zgRpmqqVlqHvPEXp8iQZxFp2v8wUMUgdG25dX0/1SQWMmTAx74ljPjXoQFVbG1vUYRqPBInC+iE4Vjd1bjBBnBk0VzF9xZd4uIUMW9RlIQV/yq5o/Tz6DyQ0EVX7OYqy7/j7Fcp30FzoxUAjQ50ZCiihvYmgAAaQUMvVAM1sqv0x0S1LLOs5bQn/T0vIuAvbdP3aBfbPbgz60X6FB82v1ILHqTOBOBgc11gSYGK91M2LgwNXY7XbbxYpPkkw2V/R81k4Q0spjBqZ73RSUwZiqvDR8ITWmIArZ0S
C/kijq7R1r4YqPaIBuD02Jxs0YNqnfCTJ5s1mUXxZ1+nBYoAM0Wt14W2cV5tXbfz2H6K6mjAoUFJoLGa6wU1m+48Gqyr5116mL46KfAMHaq/bHqRCDArgYb2Y18CBTrTt8C+kRAaKalgnN2Epeo+X9zEd3+82oTq9XccBYtzJv+sJrClPw0240xKBlkRVFArw9pRE8UXeys/cA==

STEP 5 Decrypt <Keyword>:

Clean Signature

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments" />
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1" />
    <Reference URI="">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>QHlmDFy3n2yowTKjR6PIACKr4ss=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>AHLtvJV1r3E81+HHqNXzxH7RIk9JSeEh0yLe6h2xz7KLgn+vsbya+a==</SignatureValue>
  <KeyInfo>
    <KeyValue>
      <DSAKeyValue>
        <P>/KaCzo4Syrom78z3EQ5SbbB4sF7ey80etKII864WF64B81uRpH5t9jQTxeE0ImbzRMqzVD2kVG9x7d7n1kuFw=/P>
        <Q>li7dzDacuo67Jg7mtqEm2TRuOMU=</Q>
      </DSAKeyValue>
    </KeyValue>
    <Q>24Rxsnqc9E7pGknFFH2xqar0RPBaQ01khPMdLRQnG541Awtx/XpAf5Bpsy4pNWMcHbN0NogpsQS5Qvn1Mpa=/G>
    <Y>1fOXRxcVjboo2avgjdpU5KoBUyN1c2NPgORKrS6EtTP2Kd10BLfkdFZEL1yqSCqEFMc7WS9eFLwNZqKoA==</Y>
  </KeyInfo>
</Signature>

STEP 6 Validation Signature:

Signature passed core validation
2018-01-16
18:07:31
STEP 7 Generate Request:

Request:

```xml
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
CombinedDecision="false" ReturnPolicyIdList="false">
<Attributes Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
  <Attribute AttributeId="department" IncludeInResult="false">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">trading</AttributeValue>
  </Attribute>
  <Attribute AttributeId="http://wso2.org/claims/role" IncludeInResult="false">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">senior</AttributeValue>
  </Attribute>
  <Attribute AttributeId="year-entrance" IncludeInResult="false">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">2009</AttributeValue>
  </Attribute>
  <Attribute AttributeId="system-attempts" IncludeInResult="false">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">16</AttributeValue>
  </Attribute>
  <Attribute AttributeId="http://wso2.org/claims/age" IncludeInResult="false">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">28</AttributeValue>
  </Attribute>
</Attributes>

<Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" IncludeInResult="false">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">2</AttributeValue>
  </Attribute>
</Attributes>

<Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment">
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-date" IncludeInResult="false">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#date">2018-01-16</AttributeValue>
  </Attribute>
  <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time" IncludeInResult="false">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">18:07:31</AttributeValue>
  </Attribute>
</Attributes>
</Request>
```
STEP 8 Sending Policy to Balana:

STEP 9 Sending Request to Balana:

STEP 10 Balana Response:

Response:
<Response xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17">
  <Result>
    <Decision>Permit</Decision>
    <Status><StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/></Status>
  </Result>
</Response>

STEP 11 Output Balana:

Final Decision: Permit

C.2 Example B

STEP 1 Metadata Extraction

METADATA :
<xml version="1.0" encoding="UTF-8" standalone="yes"/>
<ImageDescription xmlns="JPSearch:schema:coremetadata">
  <CreationDate>2017-11-20T00:12:51.148+01:00</CreationDate>
  <ModifiedDate>2017-11-20T00:12:51.149+01:00</ModifiedDate>
</ImageDescription>
</ActualRightsDescription>
STEP 3 Decrypt <RightsDescription>

Clean Policy:

<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17">
  <PolicyId>Policy4_TFG</PolicyId>
  <RuleCombiningAlgId>
  </RuleCombiningAlgId>
  <Target/>
  <Rule Effect="Permit" RuleId="rule1">
    <Condition/>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-one-and-only">
          <AttributeDesignator AttributeId="http://wso2.org/claims/age" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#integer" MustBePresent="true"/>
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">26</AttributeValue>
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-less-than">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:integer-one-and-only">
            <AttributeDesignator AttributeId="system-attempts" Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#integer" MustBePresent="true"/>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">2</AttributeValue>
          </Apply>
        </Apply>
      </Apply>
    </Apply>
  </Rule>
</Policy>

58
STEP 4 Encrypted <Keyword>:

Signature encrypted:
05dxywYWkrvyt09xbD5AvI1TOMlbv0jIv7s8Xehv9cYjXrCndCnSuU4aHmhTq+N1JSOE+e0Bi
aepskgq8f8pL5oC0ch4ASu6asDydasAxF01R9Z+6cEsUD1EYI1sAIUMLdBeWyRupXQ/2hp
a3dWoQ9VPBo9e1JFAOFmg9P9x0z/KZWUE11NLJ3axP7+vxLpjk9pxZRoq1I0koB0x6GR
TxvD5QKOuxsIJJT539DxQfCNUb9kC3w35FzbdPctip7EpnJZXI3bu3VcdjWwAgpBrtSmjHD
GGwawvqrUHNgpn0UF5TnFuILqRzzsRg/Ku8Q6cFdg+4ATAinpp2sRdn6+YAM1BdCGiG2p2jW
h3e3/g8Ypq5/plcGB/fqPj+0PVWYMAM6sLyroPAIaLubuBYXXGMQkboX1hbnovhPVQIFO
EJEREOsTNDyvheEBVGHKP+wTlkyElAznVQaNob1+nuYTH4R3D3/8qAp9Tsdm91QRFCtTM
1McimWx13BFMC1XZK6m5o9l9r+be+ekXMA41CTaw1Cjt1RLoAYEJMr5JAm4dJKtrQDgC
5C1IvISS66z6FWhyjAssAxOIEcEd/hmAvYJTsEsoOF/BoYyi6456z/net/4de0oyNFHMJdhp/nX
2FyMc4zwPAB2z1ty1C0Go/ejQt8+qIDtHuws0cm13H09wn0FWy3ZpHrRfKKZ82ZugEF/Gf
uSMnpCXXS13L78Buuij3j6KCP3C3HAgYFTVjsWAAkuU5d1fBi/2458s7p7ggFz6ee4fjvb6v
iq7Qi1IBi+DylwoAYrZLl4etJr1S5JNo9exijUIRK10koM4+IYXw6y6L4sBZ/RckbyqvFvMz
XGwDacwFr7vqL7vqSxR1746Bk7ZwnyY7ZnXqszt1CjKHU03YGM1Kx4zGpqKpxvqlHg
vPEX8pi0ZXFp2v8VMUGw+d25d0X/1SOQYeMTzgTAX74jPjX0PQVbG1vUyVRgPBINcc+i4Vd1
bjbBnnB0Kv0zF9x2K4auJGw9L9QLy/uygq/TzX6Dy0x0Eyv7QVqy/7jFCp30FoucAJqo5G2i
hVYMGga0v5MCvAMISxvq0Q6L05oBq0n/70yIunAvbdPsabFbBbgz60X6FB8z1HIqO8BoB
gcnlqS9GK19M2LqXnXYBxbPpkz2W/R81k40spBqZ3R9USUw2gViDr8IT1TSpJ51M1
NgQYK2MybqvQu3Yly1N75KYE1mapM38Fr13nn0/Dhj7jTW/LoIFw8dDyq2V3ry915VlAc
y8Q9plynkkJy++TqePBpB0rqMTH1vN9gtfAqecMTszEOcYb2Y18CBcrT8C+kRaaKalgN2
Erepo+x9zEd3+82Oy3XCCBYtzjv+sJRCipw0240XBkRVFArw9pe8UXkeys/cA==

STEP 5 Decrypt <Keyword>:

Clean Signature
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<body>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod
      Algorithm="http://www.w3.org/TR/2001/REC-xmml-14n-20010315#WithComments"/>
    <SignatureMethod
      Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
    <Reference URI=""/>
      </Reference>
    <SignatureValue>QHlmDFy3n2yowTKjR6P1ACKr4ss==</SignatureValue>
  </SignedInfo>
</Signature>
<KeyValue>
  <DSAKeyValue>
    <P>KaCzo4Syrom78z3EQ5Sbb4sF7ey80etKII864WF64B81uRpH5t9jQTxeEu0ImbzRMqzVD2kVG9xD7nN1kuFw==</P>
    <Q>li7dzDacuo67Jg7mtqEm2TRuOMU==</Q>
  </DSAKeyValue>
</KeyValue>

STEP 6 Validation Signature:
Signature passed core validation
2018-01-16
18:23:31

STEP 7 Generate Request:

Request:
  <Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
    CombinedDecision="false" ReturnPolicyIdList="false">
    <Attributes Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
      <Attribute AttributeId="department" IncludeInResult="false">
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">finance</AttributeValue>
      </Attribute>
      <Attribute AttributeId="http://wso2.org/claims/role" IncludeInResult="false">
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">senior</AttributeValue>
      </Attribute>
      <Attribute AttributeId="year-entrance" IncludeInResult="false">
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">2001</AttributeValue>
      </Attribute>
      <Attribute AttributeId="system-attempts" IncludeInResult="false">
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">2</AttributeValue>
      </Attribute>
      <Attribute AttributeId="http://wso2.org/claims/age" IncludeInResult="false">
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">36</AttributeValue>
      </Attribute>
    </Attributes>
  </Request>
STEP 8 Sending Policy to Balana:

STEP 9 Sending Request to Balana:

STEP 10 Balana Response:
Response:
<Response xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17">
  <Result>
    <Decision>Deny</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>

STEP 11 Output Balana:

Final Decision: Deny