Cloud platforms Lead to Open and Universal access for people with Disabilities and for All

---

D104.3. Security, privacy and ethical policy assurance Gateway

Project Acronym **Cloud4all**
Grant Agreement Number **FP7-289016**

Authors **Barcelona Digital Technology Centre**
Status **First Draft**
Dissemination Level **Consortium**
Delivery Date **10/11/2013**
Number of Pages **12**

Funded by EU's FP7 Led by Technosite Grupo Fundosa

Keyword List

Version History

Table 1. Version history

Revision	Date	Author	Organization	Description
1	07/25/2012	Marcel Malet	BDigital	First draft

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Table of contents

List of Tables

List of Figures

1 Initial concepts

A security gateway is a solution for enforcing identity and security for SOAP, XML, and REST based web services. It is a dedicated application which allows for a more centralized approach to security and identity enforcement, similar to how a protocol firewall is deployed at the perimeter of a network for centralized access control at the connection and port level.

1.1 Why a Security Gateway?

Cloud4all needs a component where security and identity can be enforced. Specifically, we need a component where users can be authorized and get their preferences when a concrete application, device, or network wants to use these preferences, in a specific moment or context.

2 Frameworks

In order to develop the Security Gateway we need a framework to intercept all requests and authorize them based on security policies defined by users or the system itself. At the same time we need to monitor all this activity for control and audit purposes. Therefore, we have to find a framework that can provide a base to create the gateway.

2.1 WSO2 Platform

WSO2 is an open source application development software company focused on providing service-oriented architecture (SOA) solutions for professional developers.

WSO2 Carbon is an SOA middleware platform from WSO2. Built on OSGi, Carbon encapsulates major SOA functionality such as data services, business process management, ESB routing/transformation, rules, security, throttling, caching, logging and monitoring. WSO2 products such as Application Server, Enterprise Service Bus, and Business Process Server are built on top of the WSO2 Carbon middleware platform. The WSO2 products are available to download, or can be custom-built by adding the components on top of the Carbon core **[3]**.

- **WSO2 Enterprise Service Bus (ESB):** is a simple, lightweight and high performance enterprise service bus (ESB), based on the Apache Synapse enterprise service bus, providing enhanced management and development/configuration support and SOA Governance capabilities. Ebay uses WSO2 Enterprise Service Bus as one of the key elements in its transaction software, which continuously executes $2,000 worth of transactions per second.

- **WSO Identity Server (IS):** is an open source identity & entitlement management server having support for Information Cards, OpenID and XACML.

- **WSO2 Business Activity Monitor (BAM):** is a Business Activity Monitor designed to monitor and understand business activities within a SOA deployment, which also can be extended to cater for other general monitoring requirements.

2.2 Sopera

The SOPERA Advanced Service Factory (ASF) is a sophisticated all-inclusive platform for service-oriented integration projects. Thanks to a significant technological advantage, SOPERA provides leading functionality and is specifically designed to meet practice needs. SOPERA also features strong support for service development and business process management **[1]**.

SOPERA ASF offers integrated security and identity management functionality that can be tailored to requirements. As a result, scaled security concepts can be implemented easily with the desired level of granularity. Also supports the authentication and authorization of service call-ups and permits service-message coding and signing according to specific need.

2.3 OpenAM

OpenAM provides open source Authentication, Authorization, Entitlement and Federation software. OpenAM provides core identity services to simplify the implementation of transparent single sign-on (SSO) as a security component in a network infrastructure. OpenAM provides the foundation for integrating diverse web applications that might typically operate against a disparate set of identity repositories and are hosted on a variety of platforms such as web and application servers **[2]**.

2.4 Why WSO2?

There are alternatives to the WSO2 framework. However, under our study none of them is as complete as the WSO2 middleware. The framework has to be **modular**, and **ready to be deployed in the cloud**. Another important issue is the scalability. WSO2 not only has products to enforce security and privacy but it has other products in order to make scalable the former ones. Specifically WSO2 has a product called Elastic Load Balancer, described below in this document (section 5.1).

WSO2 is a **complete middleware**, where can be found products for security and privacy, for monitoring and for scalability in the same platform. Other frameworks only have a subset of all needed components.

Another important point for WSO2 option is the big developers' **community**. There are an important number of webs and foros where can be found tutorials and solutions. At the same time, WSO2 Company **releases new versions very fast**, improving permanently their products.

Finally, the last point is the **performance**. WSO2 products are ready to manage a high amount of data. For example, as mentioned before, **Ebay** uses WSO2 Enterprise Service

Bus as one of the key elements in its transaction software, which continuously executes $2,000 worth of transactions per second.

3 Functional design

3.1 Actors and System components

1. The User: A standard Cloud4all user, with one several disabilities.
2. The Proxy Server: Component responsible for managing all traffic across the gateway. It manages the incoming and outcoming packets, sends data to monitoring system and acts as a policy enforcement point, so it applies the decision of Authorization server for a specific incoming request.
3. The Authorization Server: It is the authorization server. It has a repository of security policies and has an engine to evaluate these policies based on request form Proxy Server.
4. Activity Monitor: This component collects all information needed for audit purposes from the central point (Proxy Server).

3.2 Use Case diagram

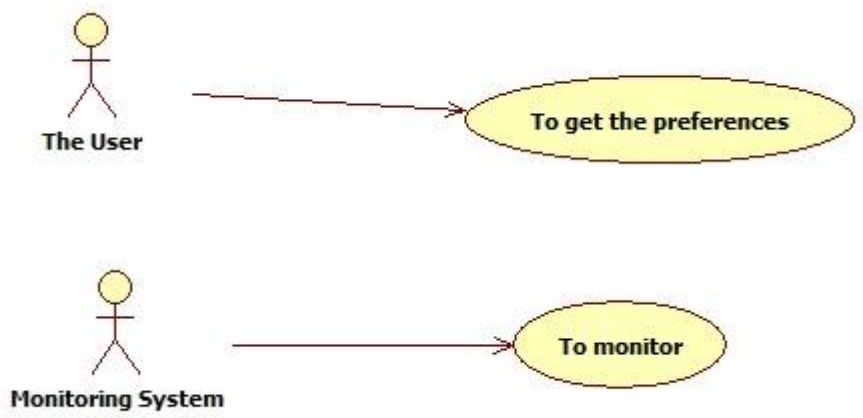

Figure 1. Use case diagram

3.3 Sequence diagram

Use case. To get the preferences:

- Description: The user wants to recover his preferences from the Preferences Server in the cloud.
- Actors: User, Proxy Server, Authorization Server, Preferences Server.

- Preconditions: The user has to have his user preferences in the Preferences Server
- Postconditions: ----

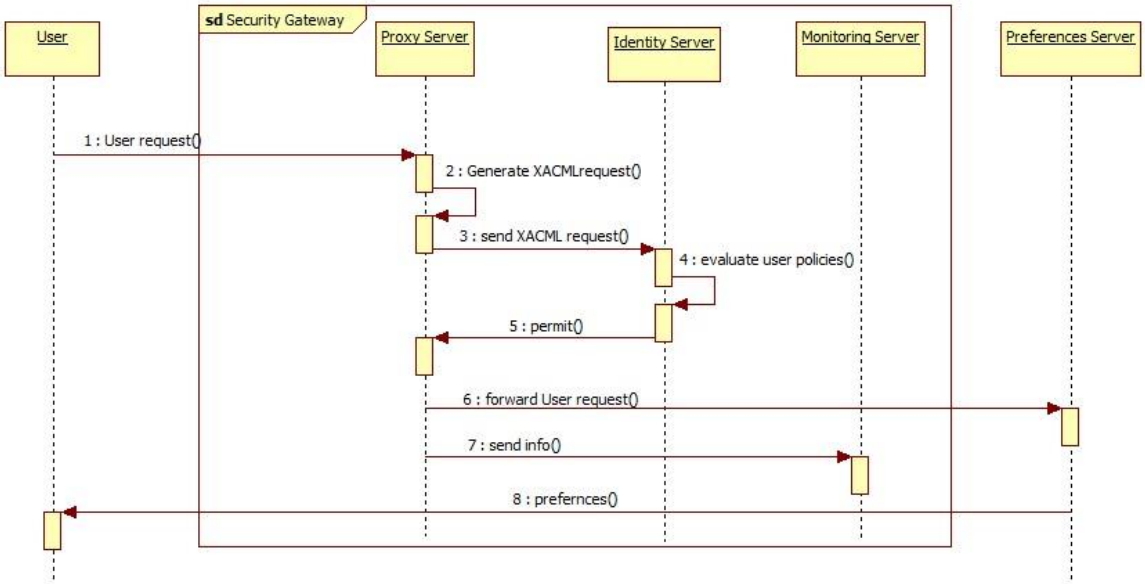

Figure 2. Sequence diagram

4 Technical design

4.1 General overview

The security gateway is composed by 3 elements. The Proxy server, which is the central point of the gateway. The Authorization server which acts as a decision point and where all security policies are stored and, finally the Monitoring system, which records all the activity for log and audit purposes.

4.2 Components description:

- **Proxy Server**: Proxy server is based on the WSO2 Enterprise Service Bus (WSO ESB). All traffic will cross this component. It permits the development of custom proxies in order to apply a set of processes on traffic crossing across. Therefore, it permits to ask for authorization to other component and, at the same, time send data to a monitoring system.

- **Authorization Server**: The Authorization server is based on the WSO2 Identity Server (WSO IS). It is a Policy Decision Point (PDP), so where, based on a request,

authorization decisions are made. So, it has a repository of all security policies and where these policies are evaluated.

- **Monitoring system**: This component also based on WSO2 Business Activity Monitor (WSO2 BAM) is responsible of record all relevant data crossing the gateway.

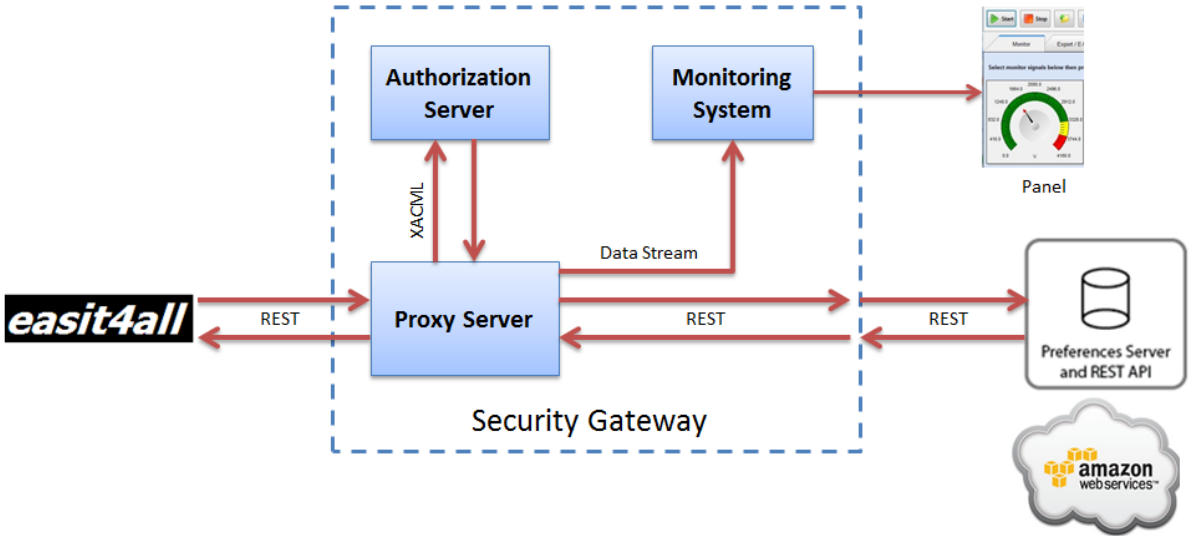

Figure 3. Security Gateway Architecture

4.3 XACML Policies

Security policies are defined with XACML. XACML (eXtensible Access Control Markup Language) defines a declarative access control policy language implemented in XML and a processing model describing how to evaluate access requests according to the rules defined in policies.

XACML policies can have several rules for different purposes. Basically, all rules have 4 attributes.

1. The **Resource**: It is the resource or service which we have to protect the access. In our case the "Preferences Server".
2. The **Action**: What we want to do over the resource. In our case we want to "read" the user preferences.
3. The **Subject**: It is the entity which wants to get the data. In our case the "user token".
4. The **Environment**: This is a set of additional customizable attributes. We have created 2 new attributes, the **Application ID**, and the **Origin IP**. We could add another one, the **device type** or **device ID**.

With all this parameters the gateway will decide if the request is permitted or denied.

5 Scalability

We can replicate the entire gateway, using a load balancing module. The next image illustrates how.

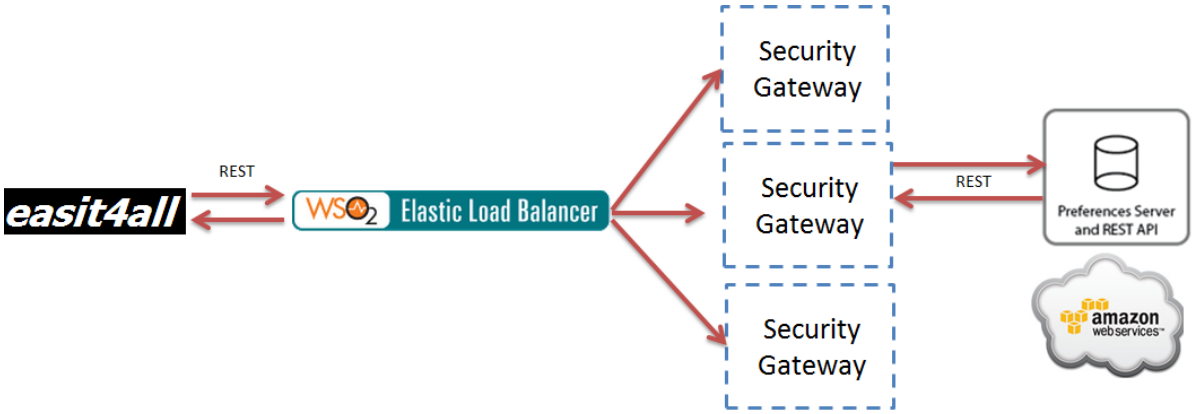

Figure 4. Load balancing scheme

5.1 WSO2 Elastic Load Balancer

The WSO2 Elastic Load Balancer (WSO2 ELB) would offer a lean approach to balance loads across the Security Gateways instances. It provides fail-over, auto-scaling, and multi-tenancy letting services scale automatically with dynamically changing load characteristics.

6 Prototype limitations

In this prototype there are two users, already configured with their security policies in the Authorization Server. The prototype doesn't have a user interface to create these security policies, so it has to be created using a source code view in the Authorization Server.

Regarding security parameters, we have used the user token, an application ID, the origin IP (the network) and also what kind of Operating System has the device, but we can use much more information such as the date, the hour, GPS data, etc…

At the same time, the prototype uses a specific parameters configuration in the GET requests that could be not exactly the same in other parts of the project. Specifically:

Http…./services/Entitlement_prova/<userToken>/settings/<appToken>/net/<localIp>/device/<device>

Very similar to:

http://preferences.gpii.net/user/<userToken>

7 References

[1] Sopera, http://www.sopera.de/en/products/sopera-features/

[2] OpenAM project, http://openam.forgerock.org/

[3] WSO2 Platform, http://wso2.com/