Design Document

**WSO2 Identity Server**

**Contact:** Jonathan Marsh  
**Email:** jonathan@wso2.com  
**Tel:** 916 663 2001

**Address:** 4131 El Camino Real, Suite 200,  
Palo Alto, CA 94306, USA  
**Tel:** +1 408 754 7388  
**Fax:** +1 408 689 4328  
**Email:** bizdev@wso2.com
Table of Contents

1.0. Introduction

2.0. System Overview
   2.1. System & User Identity Management
   2.2. Entitlement Management
   2.3. XACML 2.0/3.0 Support
   2.4. Manage & Monitor

3.0. Design Considerations
   3.1. Operating Environment
      3.1.1. Hardware Requirements
      3.1.2. Software Requirements
   3.2. End-User Characteristics
   3.3. Goals and Guide-lines
   3.4. Development Process

4.0. System Architecture
   4.1. Abstract System View

5.0. System Design
   5.1. High Level Design
   5.2. Use Cases
   5.3. Deployment Diagrams
   5.4. Sample User Interfaces
1.0. Introduction

The WSO2 Identity Server is a product providing security and identity management of enterprise web applications, services, and APIs, built on the WSO2 Carbon platform.

WSO2 Identity Server enables enterprise architects and developers to improve customer experience by reducing identity provisioning time, guaranteeing secure online interactions, and delivering a reduced single sign-on (SSO) environment. The WSO2 Identity Server decreases identity management and entitlement management administration burden by including role base access control (RBAC) convention, fine-grained policy based access control, and SSO bridging. WSO2 Identity Server is expected to offer significant time saving and affordable acquisition. Purpose-built for rapid configuration and efficient extension, WSO2 Identity Server is expected to be easy to configure and extend. These attributes lead to lower investment and higher ROI.

In the following chapter, System Overview will be discussed. This document will discuss about the design considerations, Overall system architecture, and the detailed system design in the latter chapters, providing the design decisions of the WSO2 Identity Server.
2.0. System Overview

WSO2 Identity Server is a product built on top of WSO2 Carbon. Based on the OSGi specification, it enables easy customization and extension through its componentized architecture. We will discuss the important features of the Identity Server in this chapter. The users are given the choice of deployment to on-premise servers, private cloud or public cloud (WSO2 StratosLive Identity-as-a-Service) without configuration changes.

2.1. System & User Identity Management

WSO2 Identity Server implements flexible user store via built-in LDAP (powered by ApacheDS), external LDAP, Microsoft Active Directory or any JDBC database. It provides an API for integrating identity management to any application. WSO2 Identity Server supports multiple profiles per user using its flexible profile management feature.

WSO2 Identity Server enables Single Sign-On (SSO) for the enterprise applications via OpenID, SAML2, and it comes bundled with Apache KDC out of the box, supporting Kerberos Key Distribution Center. It also provides an SSO bridging between on-premise systems and cloud applications, enhancing the single sign-on experience of the user. It also provides multi-factor authentication for OpenID, via XMPP.

WSO2 Identity Server implements REST security with OAuth 2.0 and XACML. WSO2 Identity Server integrates with Microsoft SharePoint with Passive STS support. It provides credential mapping across different protocols. Provisioning via SCIM instead of legacy SPML and Auditing via XDAS too are supported. Delegation is supported via OAuth 1.0a, OAuth 2.0, and WS-Trust, while Federation is supported via OpenID, SAML2 and WS-Trust STS. XKMS for key storage and distribution too is provided by the Identity Server.

2.2. Entitlement Management

WSO2 Identity Server contains an advanced entitlement auditing and management. It provides entitlement management for any REST or SOAP calls. WSO2 Identity Server provides Attribute and Claim based access control via XACML, WS-Trust, OpenID and claim management. WSO2 Identity Server also provides Role based access control (RBAC) and Fine-grained policy based access control via XACML.

2.3. XACML 2.0/3.0 Support

WSO2 Identity Server provides a friendly user interface for policy editing. It also supports multiple Policy Information Point (PIP) and policy distribution to various Policy Decision Points (PDPs). It provides a high performance network protocol (over Thrift) for PEP/PDP interaction, and policy decision and attribute caching.
Notifications are provided for policy updates. Moreover, the WSO2 Carbon TryIt tool that comes bundled with the Identity Server lets the user explore the policy impact.

2.4. Manage & Monitor

WSO2 Identity Server provides a comprehensive management console with enterprise-level security. It also comes with a built-in collection and monitoring of standard access and performance statistics. Operational audit and KPI monitoring and management is achieved by integrating with the WSO2 Business Activity Monitor. Further key metrics monitoring and management is achieved with JMX MBeans. WSO2 Identity Server offers a flexible logging support with integration to enterprise logging systems. WSO2 provides a centralized configuration management across different deployment environments with lifecycles and versioning, with integration to WSO2 Governance Registry.

3.0. Design Considerations

In this chapter, we will discuss the design time considerations that were made about the environment and the users.

3.1. Operating Environment

3.1.1. Hardware Requirements

1. Pentium 800MHz or equivalent processor
2. 256 MB of memory
3. The extracted binary distribution of WSO2 Identity Server takes up about 130 MB of disk space (WSO2 Identity Server will require more disk space at runtime to keep temporary files, server logs etc)

3.1.2. Software Requirements

1. Java Runtime Environment 1.5 or above

3.2. End-User Characteristics

WSO2 Identity Server mostly targets the enterprise level users. The system administrator, the one who has access to the server, configures the Identity Server. Identity Server can be configured using the configuration files in the conf folder. The default admin (comes in the Identity Server with the name ‘admin’), is often same as the administrator, who configures the Identity Server. More users can signup later. The admin user can add more users, and assign roles to them. The roles may reflect the actual role of the user (devops, developer, tester, lead, ..) and will have the relevant privileges assigned to them by the admin.

Admin user is expected to be able to configure the Identity Server, and the environment. Users can be anyone
who access the Identity Server’s management console. Identity Server can also be used by those who use it as an identity provider, without any knowledge of security or identity. Those users will have minimal knowledge of the Identity Server or its configuration, while consuming it as an identity solution for their applications.

3.3. Goals and Guide-lines
Goal of the Identity Server is to be a lightweight, developer Friendly, complete, and easy to deploy identity solution. WSO2 focuses to provide a complete SOAP API for integrating or for embedding into any application or system. It is expected to provide pluggable workflows for privileged operations. It should be extensible for pluggable authenticators, alternative user stores, XACML/SAML extension points and more. Identity Server provides the deployments with high availability through clustering.

3.4. Development Process
WSO2 follows Apache Process, which is similar to the Agile process, in development. As an open source project, all the architectural, design, and development are done public, discussed through the public mailing lists, and committed to the public source repository.
4.0. System Architecture

4.1. Abstract System View

Each server in the WSO2 platform is built using the Carbon platform. Carbon Server is a term used to depict any product, such as WSO2 Enterprise Service Bus, WSO2 Application Server, and WSO2 Identity Server, that is built on top of the Carbon Platform.

Figure 1: Abstract System View
5.0. System Design

5.1. High Level Design

The authentication and authorization choices in WSO2 platform, with the WSO2 Identity Server is depicted in Figure 2. Authentication needs a user store that holds the information about users and “Enforcement Point” that verifies the credentials against the User store.

Carbon Servers support two user stores:

1. Database based user store
2. LDAP based user store

It is a common deployment pattern for multiple carbon servers in a single deployments to point to the same user store, and this provide a single point to control and manage the users.

WSO2 Carbon based products can be configured to authenticate any incoming requests. WSO2 Carbon supports options including HTTP Basic Authentication over SSL for HTTP, WS-Security User Name Tokens, and
Web SAML SSO. This authentication is done against the users that reside the user store. Each Carbon server has a Web Service called Authentication Admin Web Service, which exposes the authentication as a Web Service to the outside. The client can invoke the Authentication Admin Web Service and get a HTTP Cookie after logging in and reuse the Cookie to do authenticated calls to a Carbon Server.

Authorization

In Authorization Scenarios, Carbon server receives a request that is generally already authenticated or a request that include a token. In either case, we want to check whether the authenticated user have enough permission to carry out a given action.

Three roles can be defined in such scenarios, using the XACML terminology.

1. PEP (Policy enforcement Point) intercepts requests and makes sure only authorized requests are allowed to proceed.
2. PDP (Policy definition Point) stores the permissions and verify that given user have enough permissions
3. PAP (Policy Administration Point) let users define and change permissions.

Carbon servers support the Policy Enforcement Point (PEP) role using a WSO2 ESB Mediator or Apache Axis2 Handler or through a custom user code.

![PEP using WSO2 ESB Mediator](http://wso2.com)

WSO2 Identity Server supports three ways to define permissions, for Policy Definition Point (PDP).

1. Database based permission stores – permissions are stored in the Database
2. XACML – permissions are described using XACML specification
3. KDC - Permissions are provided as Kerberos Tokens

The Identity Server supports policy administration (PAP), which enables the users to edit the permission definitions through the management console.

1. If the Database based permission store is used, Carbon Server can be configured to connect to the permission database directly and load the permissions to the memory. Then it authorizes user actions using those permission descriptions. Carbon servers also have an Authorization Admin Web Service that lets users check for permissions of a given user remotely.
2. If XACML based authorizations are used, there must be an Identity Server that acts as a PDP (Policy Definition Point). Each Carbon server (acting as the PEP, Policy Enforcement Point) invokes an Entitlement Service available in the Identity Server to check the permissions. Entitlement service is available as a Web Service or a Thrift Service.
3. If Carbon server receives a Kerberos token, it talks to a configured Kerberos Server and verifies the token.

5.2. Use Cases

The Identity Server lets the user sign in with an OpenID. The user can also enable XMPP based authentication for OpenIDs. The PIN number should be provided in the authentication process, if chosen. The user can also get an OpenID with the Identity Server, as well as download a Managed Information Card or an OpenID Information Card.

![Figure 4: Sign-in use case.](http://wso2.com)
Figure 5: XMPP-Based Authentication for OpenIDs

Enable XMPP-based authentication for OpenIDs

Figure 6: InfoCard/OpenID use case

User

Get Managed Information Card

User

Get OpenID

Get OpenID Information Card
5.3. Deployment Diagrams

Figure 7: WSO2 Identity Server in a production deployment.

WSO2 Identity Server can be deployed with multiple applications or servers, to fit the requirement. Figure 7 depicts WSO2 Identity Server in one of its common deployment patterns, where it is integrated to WSO2 Enterprise Service Bus for authorization and all WSO2 Carbon products for authentication.
5.4. Sample User Interfaces

As a product built on top of Carbon, WSO2 Identity Server uses Carbon UI, as its user interface. This ensures that the user with experience using some Carbon product start using the Identity Server, without any learning curve.

Figure 8: Identity Server Log in Page

Figure 8 shows the log in page which shows up, when the user loads identity server via a browser, once it is started. This page provides the options to sign-in with user name / password or using InfoCard/OpenID. It also provides the option to Sign-up, and manage OAuth.
Figure 9: Identity Server Home Page, depicting the admin user logged in.

Figure 9 depicts the home page view with the admin user signed in. It shows the server information, along with the menu items for each of the actions the user is authorized to perform.

Figure 10: Identity Server Home Page, depicting a user with limited privileges logged in.
Figure 10 depicts the home page of the Identity Server when a non-admin user with just log in privileges logged in. Since the user doesn’t have privileges to perform many of the operations, the respective menu items are not shown for the user.