IDENTITY MANAGEMENT FOR DEVELOPERS

Prabath Siriwardena
Senior Director - Security Architecture
prabath@wso2.com | prabath@apache.org

IAM FUNDAMENTALS

IDENTITY AND ACCESS MANAGEMENT

- Identity and Access Management (IAM) is the security discipline that enables the right individuals (or things) to access the right resources at the right times for the right reasons. (Gartner)

IDENTITY FEDERATION & SINGLE SIGN ON

OVERVIEW

- Single Sign On - login once - access a set of services without further login.
- Federated identity management enables identity information to be developed and shared among several entities and across trust domains.
- Single Sign On can be within a single trust domain and between multiple trust domains.

STANDARD BASED IDENTITY FEDERATION

- SAML 2.0 Web SSO
- OpenID Connect
- WS-Federation
- OpenID
- CAS

DEFINITIONS

- Identity Provider

  The authority behind user identities

  Makes assertions about users (authentication, authorization, attribute)

- Relying Party / Service Provider / Client

  Relying on an assertion provided by the identity provider. Provides services to end users

  Can be a mobile app / web app Trusts one or more identity providers

SAML 2.0 OVERVIEW

- An XML standard for exchanging authentication and authorization data between entities which is a product of the OASIS Security Services Technical Committee.
- SAML 1.0 was adopted as an OASIS standard in Nov 2002
- SAML 1.1 was ratified as an OASIS standard in Sept 2003
- SAML 2.0 became an OASIS standard in Mar 2005
- Liberty Alliance donated its Identity Federation Framework (ID-FF) specification to OASIS, which became the basis of the SAML 2.0 specification. Thus SAML 2.0 represents the convergence of SAML 1.1, Liberty ID-FF 1.2, and Shibboleth 1.3.

SAML 2.0 BASE STANDARDS

- Extensible Markup Language (XML)
- XML Schema
- XML Signature
- XML Encryption (SAML 2.0 only)
- Hypertext Transfer Protocol (HTTP)
- SOAP

SAML 2.0 COMPONENTS

- Assertions

  Authentication, Attribute and Authorization information
- Protocol

  Request and Response elements for packaging assertions
- Bindings

  How SAML Protocols map onto standard messaging or communication protocols
- Profiles

  How SAML protocols, bindings and assertions combine to support a defined use case

SAML 2.0 WEB SSO WITH SALESFORCE

- Create an account at https://developer.salesforce.com
- Add WSO2 Identity Server as a trusted identity provider to Salesforce
- Add Salesforce as a trusted service provider for WSO2 Identity Server
- Configure service provider requested claims
- Configure service provider claim mappings
- Configure role mappings
- View SAML request/response using SSO tracer firefox plugin

ENABLE MFA WITH FIDO FOR SALESFORCE

- Register a FIDO device against the user account.
- Enable FIDO as the 2nd factor for Salesforce service provider in WSO2 IS.

ENABLE MFA WITH OTP FOR SALESFORCE

- Configure Twilio as the OTP provider.
- Enable OTP as the 2nd factor for Salesforce service provider in WSO2 IS.

SAML 2.0 WEB SSO WITH GOOGLE APPS

- Add WSO2 Identity Server as a trusted identity provider to Google Apps
- Add Google Apps as a trusted service provider for WSO2 Identity Server
- Configure service provider requested claims
- Configure service provider claim mappings
- Configure role mappings
- View SAML request/response using SSO tracer firefox plugin

SAML 2.0 WEB SSO WITH TOMCAT

- Enable TLS in Tomcat
- Deploy a sample web app and enable SAML SSO filter
- Configure WSO2 IS as a trusted identity provider in Tomcat web app
- Configure Tomcat web app as a trusted service provider
- Mandatory required attributes
- Demo SAML logout

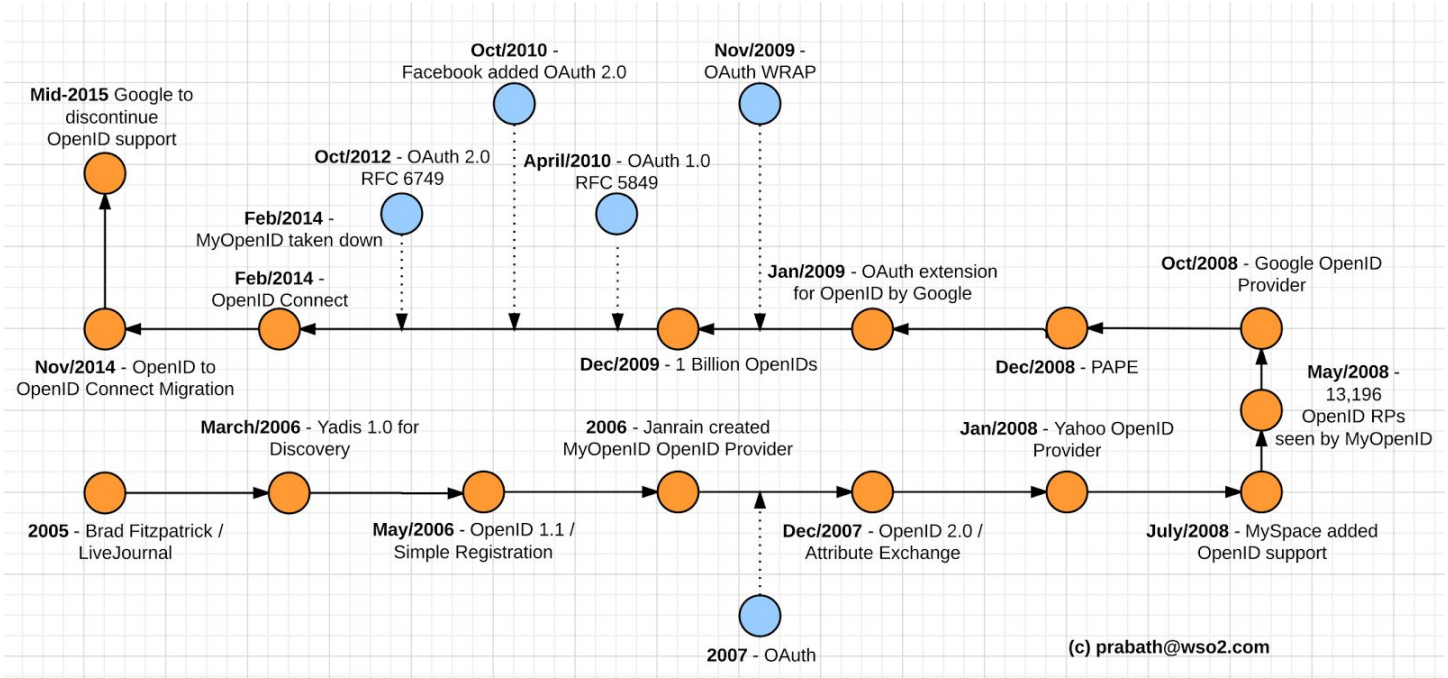
LOGIN WITH FACEBOOK

- Create a Facebook App https://developers.facebook.com/apps
- Register Facebook as a trusted federated identity provider in WSO2 IS.
- Identity provider claim mapping
- Enable Facebook login for the service provider corresponding to the Tomcat web app.

HOME REALM DISCOVERY

- <TODO>

OPENID CONNECT



SAML 2.0 vs. OPENID CONNECT

- SAML is XML based while OIDC is JSON based
- SAML has multiple bindings while OIDC has one binding
- Both are enterprise ready In last couple of years - there were more OIDC implementations than SAML
- OIDC is more mobile and SPA friendly
- Build a new app today? Use OIDC!

OPENID CONNECT WITH TOMCAT

- Enable TLS in Tomcat
- Configure WSO2 IS as a trusted identity provider in Tomcat web app
- Configure Tomcat web app as a trusted service provider with OpenID Connect
- Configure requested claims and different scope values

OPENID CONNECT WITH cURL

- https://localhost:9443/oauth2/authorize?client_id=CLIENT_ID&scope=openid&redirect_uri=REDIRECT_URI&response_type=code
- curl -k --user "$CLIENTID:$CLIENTSECRET" -d "code=$CODE&grant_type=authorization_code&client_id=$CLIENTID&redirect_uri=$REDIRECTURI" https://localhost:9443/oauth2/token

REQUEST PATH AUTHENTICATION WITH OIDC

- <TODO>

OPENID CONNECT SESSION MANAGEMENT

- <TODO>

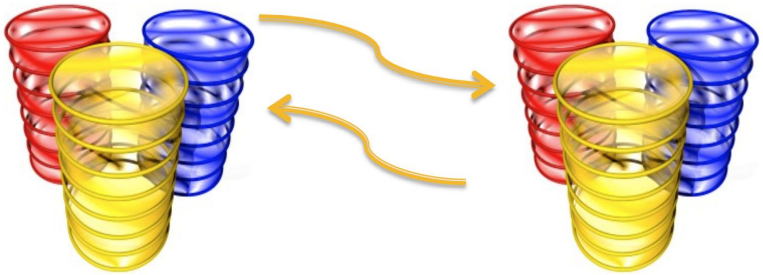
PKCE WITH OPENID CONNECT

- <TODO>

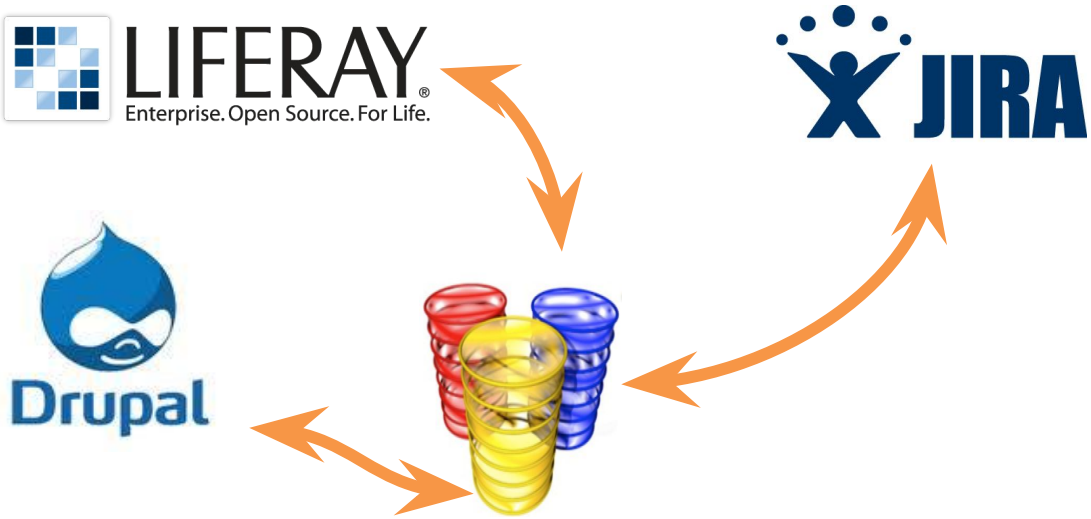
IDENTITY PROVISIONING

SYNCHRONIZATION

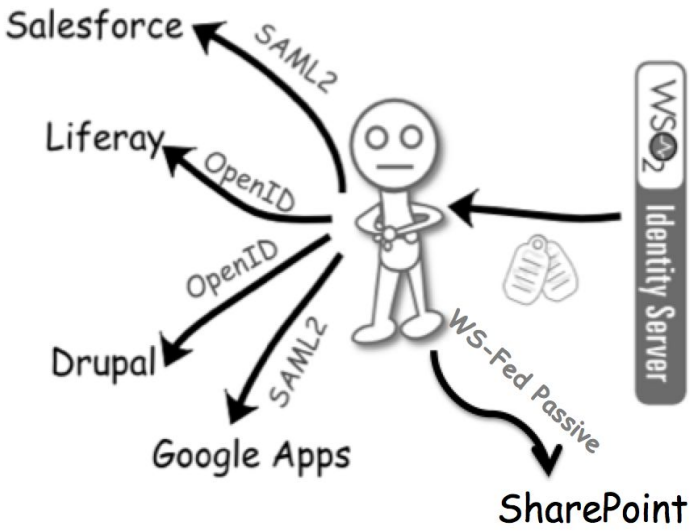
SYNCHRONIZATION



SHARING



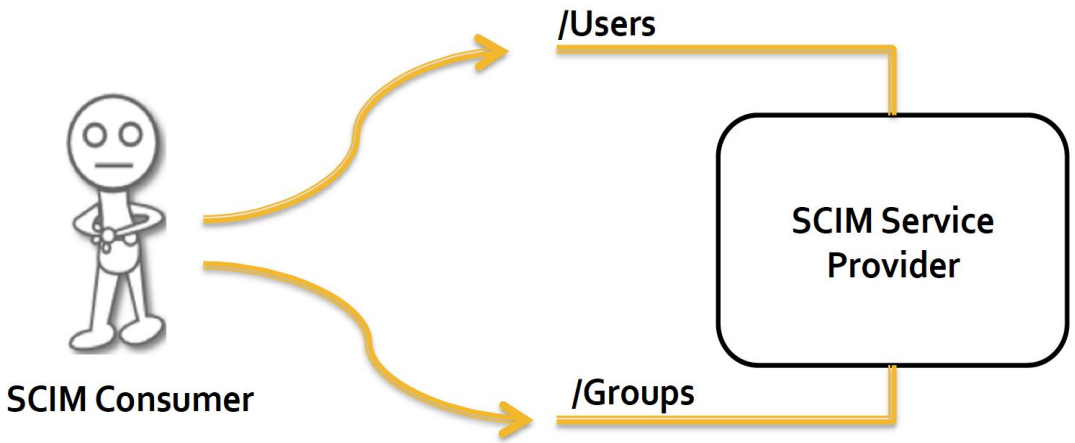
SINGLE SIGN ON



PROVISIONING

STANDARD BASED PROVISIONING

SCIM 1.1



SCIM 1.1

```
{"schemas":[],
 "name": {"familyName":"siriwardena",
          "givenName":"prabath"},
 "userName":"prabath",
 "password":"prabath123",
 "externalId":"prabathext",
 "emails":[ {"primary":true,
          "value":"prabath@wso2.com",
          "type":"home"},
         {"value":"prabathsiriwardena@yahoo.com",
          "type":"work"}]
}
```

curl -k --user admin:admin -d @add-user.json --header "Content-Type:application/json" https://localhost:9443/wso2/scim/Users

SCIM 1.1

```
{
  "schemas":["urn:scim:schemas:core:1.0"],
 "displayName" : "OSDC",
 "externalId" : "OSDC",
 "members": [
   {
     "value": "f64e6507-756d-4a14-ac43-c9d02167f411",
     "display": "prabath"
   }
 ]
}
```

curl -k --user admin:admin -d @add-group.json --header "Content-Type:application/json" https://localhost:9445/wso2/scim/Groups

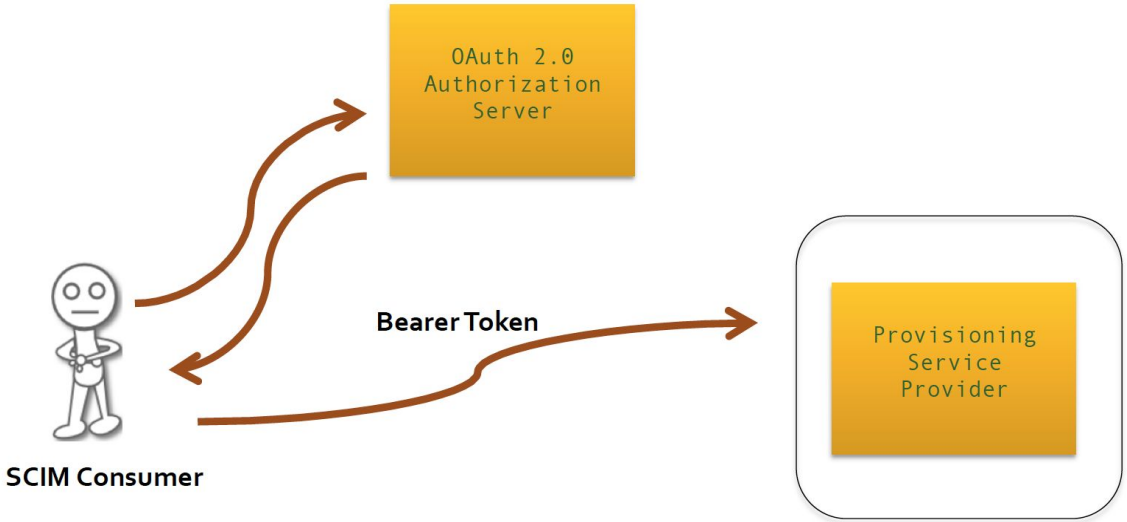
SCIM 1.1

Resource	Endpoint	Operations	Description
User	/Users	**GET**, **POST**, **PUT**, **PATCH**, **DELETE**	Retrieve/Add/Modify Users
Group	/Groups	**GET**, **POST**, **PUT**, **PATCH**, **DELETE**	Retrieve/Add/Modify Groups
Service Provider Configuration	/ServiceProviderConfigs	**GET**	Retrieve the Service Provider's Configuration
Schema	/Schemas	**GET**	Retrieve a Resource's Schema
Bulk	/Bulk	**POST**	Bulk modify Resources

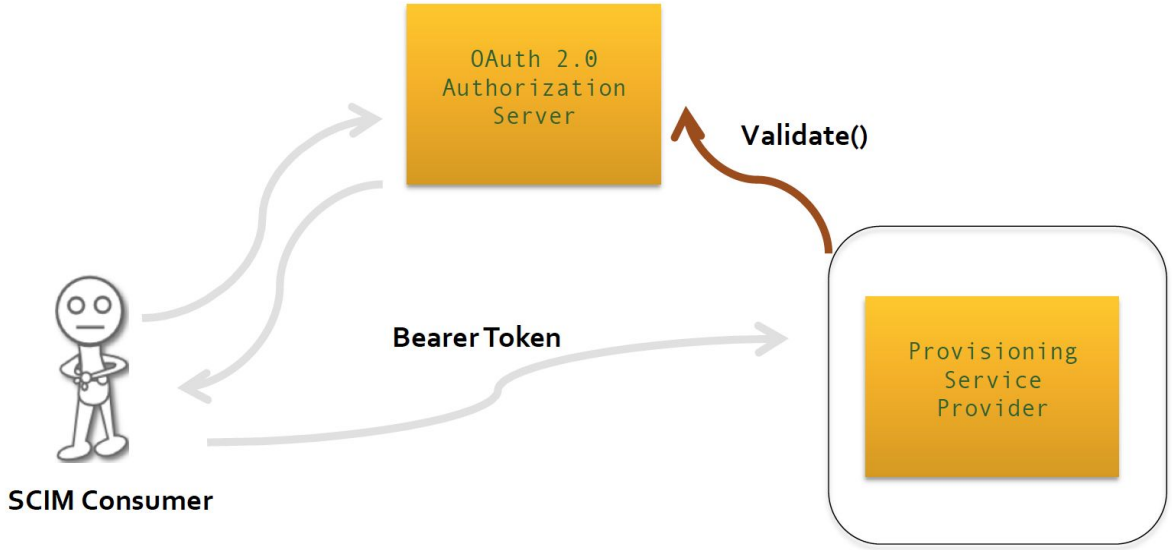
AUTHENTICATING SCIM REQUESTS

- HTTP Basic Authentication
- OAuth 2.0

AUTHENTICATING SCIM REQUESTS



AUTHENTICATING SCIM REQUESTS



CREATE USERS WITH SCIM

- Create an access token

  curl -v -X POST --basic -u $CLIENTID:$CLIENTSECRET -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -k -d "grant_type=client_credentials" https://localhost:9443/oauth2/token

- Create user

  curl -k -H "Authorization: Bearer $TOKEN" -d @user.json --header "Content-Type:application/json" https://localhost:9443/wso2/scim/Users

CREATE GROUPS WITH SCIM

- Create an access token

  curl -v -X POST --basic -u $CLIENTID:$CLIENTSECRET -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -k -d "grant_type=client_credentials" https://localhost:9443/oauth2/token

- Create user

  curl -k -H "Authorization: Bearer $TOKEN" -d @group.json --header "Content-Type:application/json" https://localhost:9443/wso2/scim/Groups

OUTBOUND PROVISIONING WITH SCIM

- Spin up another WSO2 IS instance on port 9445.
- Configure SCIM outbound provisioning connector in the 1st WSO2 IS
- Engage SCIM outbound provisioning connector for the Resident Service Provider.

JIT PROVISIONING

- Enable JIT provisioning for Facebook identity provider.
- Login with Facebook

JIT PROVISIONING + OUTBOUND PROVISIONING

- Enable JIT provisioning for Facebook identity provider
- Login with Facebook
- Enable outbound provisioning for the corresponding service provider

ACCESS CONTROL

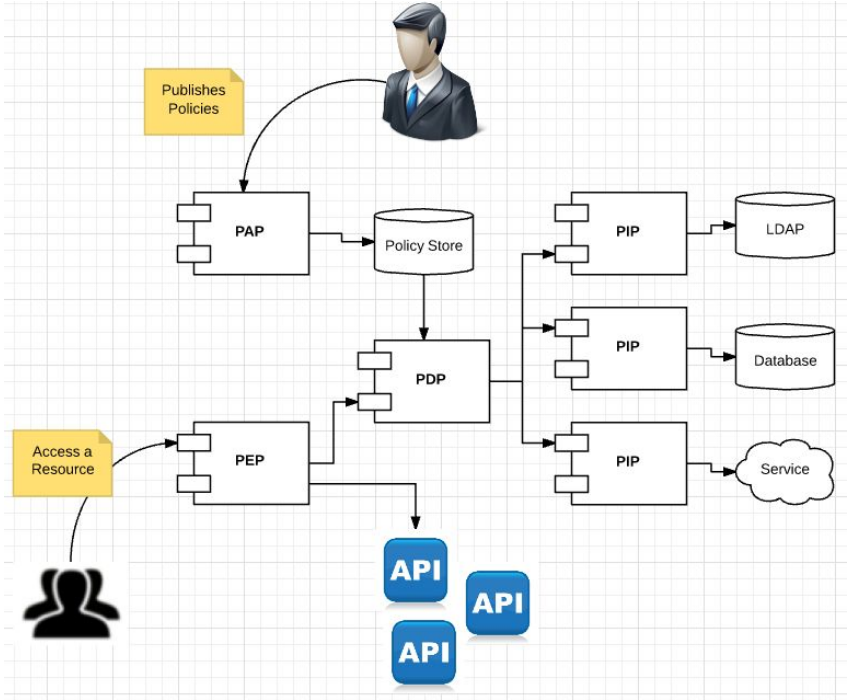
OVERVIEW

- Permission : A capability / Negative permissions are hard
- Role : A set of permissions
- Group : A set of users
- Role Based Access Control (RBAC) : Make access control decisions based on roles
- Attribute Based Access Control (ABAC) : Make access control decisions based on attributes

XACML OVERVIEW

- Fine-grained Access Control
- Requirements from Health Care, DRM, Registry, Financial, Online Web
- XACML 1.0 - OASIS Standard – 6 February 2003
- XACML 1.1 – Committee Specification – 7th August 2003
- XACML 2.0 – OASIS Standard – 1 February 2005
- XACML 3.0 – OASIS Standard – 10th Aug 2010

XACML REFERENCE ARCHITECTURE



XACML POLICY LANGUAGE

- XML based
- Represents access control logic in rules
- A given XACML policy can have multiple rules
- A XACML engine can have multiple XACML policies
- Only the XACML policies applicable to a given XACML request will be evaluated.

XACML POLICY LANGUAGE

- The smallest execution unit in a XACML policy is a Rule
- A Rule can return back Permit or Deny
- Rule combining algorithms decide how to combine multiple decisions from multiple Rules
- The policy combining algorithms decide how to combine multiple decisions from multiple policies.
- Obligations and Advices

XACML REQUEST/RESPONSE PROTOCOL

- The XACML core specification defines XML based schema for the XACML request and response.
- JSON Profile for XACML define XACML request and response in JSON
- The REST profile XACML define how to invoke the XACML PDP in a RESTful manner.
- Multiple decisions

HANDS ON

XACML EDITOR + TRY IT

- Export XACML policies
- Edit using the UI
- TryIt
- Deploy

XACML REST API

- curl -k --basic -u $USERNAME:$PASSWORD -d @xacml.json --header "Content-Type:application/json" https://localhost:9443/api/identity/entitlement/decision/pdp

CONDITIONAL AUTHENTICATION

- Pick the corresponding policy template, edit and deploy
- Enable authorization for the corresponding service provider

THANK YOU

wso2.com