Security Guidelines

Version: 0.5
Date: May 31, 2017
Author: MedBiquitous Technical Steering Committee
Contact: Prasad Chodavarapu
prasad@theabfm.org
Joel Farrell joelf@us.ibm.com

Version History

Version No.	Date	Changed By	Changes Made
0.4	31 May 2017	Prasad Chodavarapu	
0.5	18 Jun 2017	Prasad Chodavarapu	Section 3.2 update on OAuth 2.0 protocol details; Section 4 added diagrams and updated recommendations; Section 5 updates on SAML2.0 usage.

MedBiquitous Standards Public License and Terms of Use

MedBiquitous Standards (including schemas, specifications, guidelines, sample documents, sample code, Web services description files, and related items) are provided by the copyright holders under the following license. By obtaining, using, and or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

The Consortium hereby grants a perpetual, non-exclusive, non-transferable, license to copy, use, display, perform, modify, make derivative works of, and develop the MedBiquitous Standards for any use and without any fee or royalty, provided that you include the following on ALL copies of the MedBiquitous Standards or portions thereof, including modifications, that you make.

1. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the following notice should be used: "Copyright © [date of release] MedBiquitous Consortium. All Rights Reserved. http://www.medbiq.org"
2. Notice of any changes or modification to MedBiquitous Standards files.
3. Notice that any user is bound by the terms of this license and reference to the full text of this license in a location viewable to users of the redistributed or derivative work.

In the event that the licensee modifies any part of the MedBiquitous Standards, it will not then represent to the public, through any act or omission, that the resulting modification is an official specification of the MedBiquitous Consortium unless and until such modification is officially adopted.

THE CONSORTIUM MAKES NO WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, WITH RESPECT TO ANY COMPUTER CODE, INCLUDING SCHEMAS, SPECIFICATIONS, GUIDELINES, SAMPLE DOCUMENTS, WEB SERVICES DESCRIPTION FILES, AND RELATED ITEMS. WITHOUT LIMITING THE FOREGOING, THE CONSORTIUM DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY, EXPRESS OR IMPLIED, AGAINST INFRINGEMENT BY THE MEDBIQUITOUS STANDARDS OF ANY THIRD PARTY PATENTS, TRADEMARKS, COPYRIGHTS OR OTHER RIGHTS. THE LICENSEE AGREES THAT ALL COMPUTER CODES OR RELATED ITEMS PROVIDED SHALL BE ACCEPTED BY LICENSEE "AS IS". THUS, THE ENTIRE RISK OF NON-PERFORMANCE OF THE MEDBIQUITOUS STANDARDS RESTS WITH THE LICENSEE WHO SHALL BEAR ALL COSTS OF ANY SERVICE, REPAIR OR CORRECTION.

IN NO EVENT SHALL THE CONSORTIUM OR ITS MEMBERS BE LIABLE TO THE LICENSEE OR ANY OTHER USER FOR DAMAGES OF ANY NATURE, INCLUDING, WITHOUT LIMITATION, ANY GENERAL, DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, OR SPECIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF ANY USE OF MEDBIQUITOUS STANDARDS.

LICENSEE SHALL INDEMNIFY THE CONSORTIUM AND EACH OF ITS MEMBERS FROM ANY LOSS, CLAIM, DAMAGE OR LIABILITY (INCLUDING, WITHOUT LIMITATION, PAYMENT OF ATTORNEYS' FEES AND COURT

COSTS) ARISING OUT OF MODIFICATION OR USE OF THE MEDBIQUITOUS STANDARDS OR ANY RELATED CONTENT OR MATERIAL BY LICENSEE.

LICENSEE SHALL NOT OBTAIN OR ATTEMPT TO OBTAIN ANY PATENTS, COPYRIGHTS OR OTHER PROPRIETARY RIGHTS WITH RESPECT TO THE MEDBIQUITOUS STANDARDS.

THIS LICENSE SHALL TERMINATE AUTOMATICALLY IF LICENSEE VIOLATES ANY OF ITS TERMS AND CONDITIONS.

The name and trademarks of the MedBiquitous Consortium and its members may NOT be used in advertising or publicity pertaining to MedBiquitous Standards without specific, prior written permission. Title to copyright in MedBiquitous Standards and any associated documentation will at all times remain with the copyright holders.

Table of Contents

1 Acknowledgements

These guidelines are based on a submission from MedBiquitous Technical Steering Committee.

- Joel Farrell, Chair
- Prasad Chodavarapu, American Board of Family Medicine
- James Fiore, American Board of Surgery
- Scott Kroyer, MedHub
- Andy Rabin, CECity
- Valerie Smothers, MedBiquitous
- Luke Woodham, St. George's University of London

2 Introduction

As Web applications increasingly require integration with a variety of distributed Web resources, the need for a robust approach to managing the identity of users across organizational boundaries has emerged. The issue of centralized Identity and Access control Management (IAM) is particularly important for enabling these applications to work seamlessly, and MedBiquitous members have expressed an interest in developing a common approach to this problem. For example, when a clinician moves from a professional association Web site to the association's journal web site, there is often a need to log in with a different user ID and password. For readers, this is at least a nuisance and potentially a barrier to important information. To articulate the need for identity management, the MedBiquitous Technical Steering Committee (TSC), identified a set of real world use cases and put together suggested solutions with commercial and Open Source software to address identity management issues for organizations. The TSC has identified the Open Authorization (OAUTH) protocol, Open ID Connect and Security Assertions Markup Language (SAML) as useful building-block technologies for addressing identity management requirements. Outlined in this document are high-level guidelines for MedBiquitous members and other interested parties on how to use SAML/OAuth/OpenID to implement IAM.

2.1 Key Definitions

A common understanding of certain security-related terms is essential.

- **Identification** refers to the process of determining who someone (a user or computer, often called a **subject**) actually is.
- **Authentication** refers to the process of an individual proving that he or she is someone who has already had their identity established. Typically, authentication involves something that the user *knows* (a password or PIN), *has* (a token or key), or *is* (a biometric).
- **Authorization** is the process of establishing what an authenticated subject can be allowed to do or access.
- **Attribute** identifies certain properties of a subject.
- **Identity and Access Management** (IAM) is a collection of policies, processes and technologies for establishing and managing user identities and resource access control.
- **Federated Identity Management** allows identity information to be ported across independent security domains.
- **Single Sign-On** (SSO) allows a user to log in once with a recognized security authority and use the returned login credentials to access multiple resources.
- **Assertions** are statements about the authentication status (authentication assertion) or authorization status (authorization assertion) or attributes (attribute assertion) of a subject or end user.
- **Resource Server** *or* **Service Provider** provides access to resources that clients are trying to access information on or services to the client.
- **Identity Provider** ***or*** **Authorization Server** owns the user identities and provides identity management functions including authentication, authorization and access control.

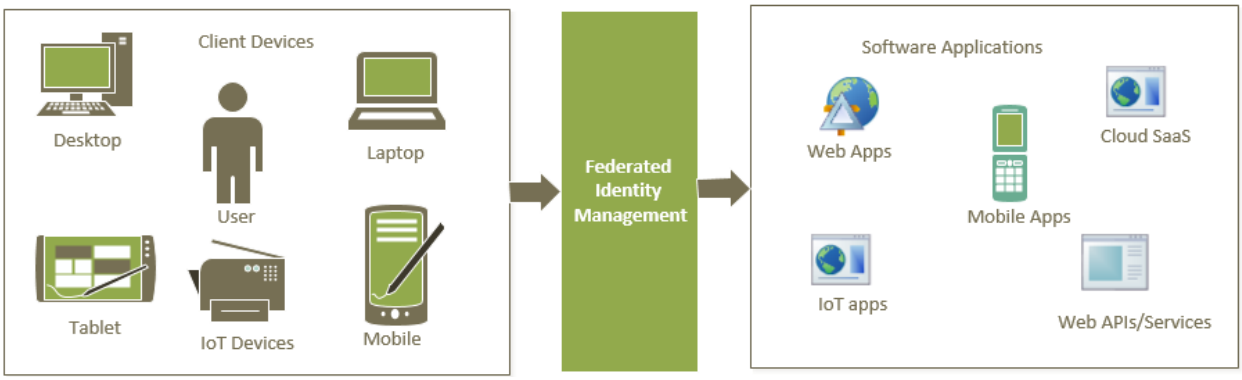
- **Resource Owner (or) End User** is the user or customer of the Client applications and who owns the resources stored on the Resource Server.
- **Client** is the applications used by an end user to interact with Resource Server. This includes web, desktop, mobile applications.
- **User Agent** is an application through which end user access the Client applications like Web Browser.
- **Native applications** are clients installed and executed on the device used by the resource owner like desktop application, native mobile application. There is no User Agent between Client and Resource Owner.
- **Authorization Grant** are credentials representing the resource owner's authorization (to access its protected resources) used by the Client to obtain an access token. Some grant types are Authorization Code, credentials, and client credentials.
- **Access Token** are credentials used to access protected resources. They include scopes, durations of access granted by the resource owner to the client.
- **Refresh Token** are credentials issued by Authorization Server to Client and used to obtain new Access Token when existing Access Token expires or when new Access tokens are required with different scope.

2.2 Why Federated Identity Management and Federated SSO

Over the past decade, we have seen dramatic changes in application landscape that resulted in different types of software applications (mobile, cloud Software as a Service (SaaS), Smart Client, Single Page Apps (SPA) and Internet of Things (IoT)). These software applications are accessed by subjects from variety of client devices including computers, laptops, netbooks and smartphones. The need for enterprises and businesses to provide a Single Sign-On (SSO) solution has greatly increased to provide end users with better usability, seamless transition between applications without re-authentication, efficient authentication processes and opportunity. Need for enterprises to support Federated SSO is also increasing to take advantage of Federated SSO solutions from identity providers like (Google, Microsoft, Salesforce and Facebook) as it will also reduce risk of enterprises from theft of identity information and reduce effort on identity information management.

As software industry moves towards cloud, the need for Federated SSO solution that works for both Enterprise deployed applications and cloud deployed applications or a combination of both becomes very important for business agility. To that end, many cloud providers have offerings for SSO solutions that provide enterprise applications an integrated identity security mechanism.

Typical federated identity management provides a centralized control to all applications and resource access from subjects using different client devices.



2.3 Cloud-based Offerings for SSO:

The TSC does not advocate any particular security solution, but list the following, and others in subsequent sections, as reasonable candidates for members to investigate.

- Ping Identity
- OneLogin
- SalesForce
- Apigee
- Microsoft Azure Active Directory
- Google Identity Platform
- Amazon AWS Directory Service
- Okta
- Symplified

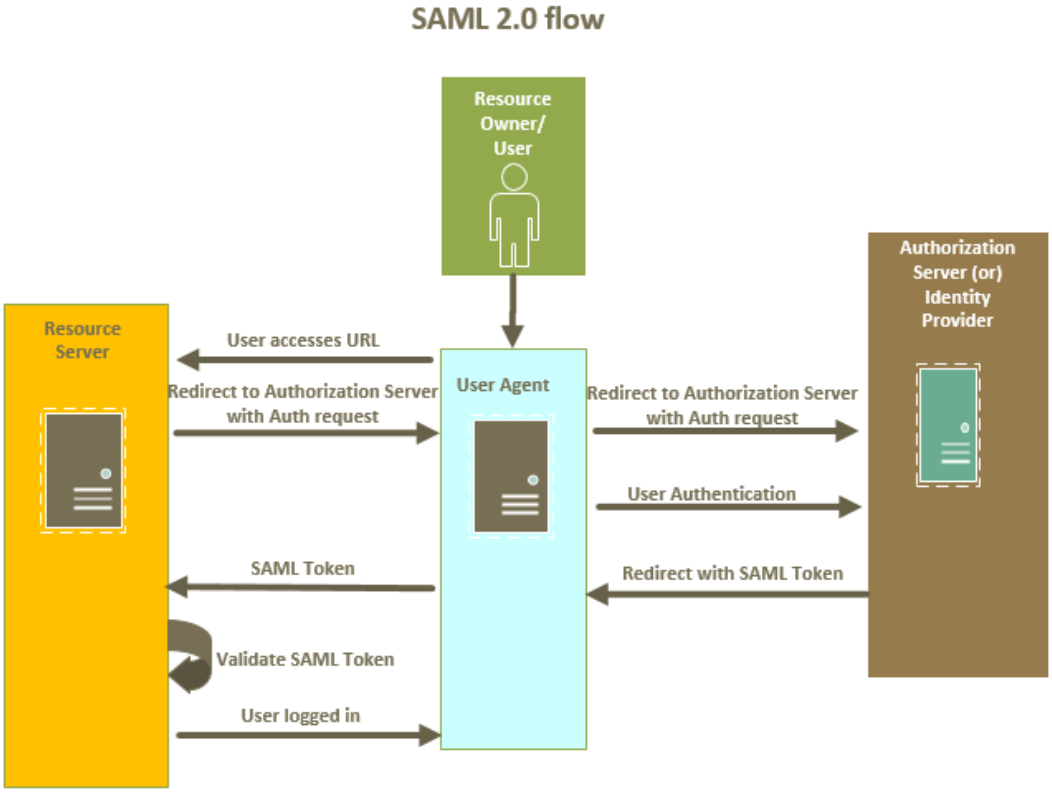
3 Security Protocols and Standards

In this section, a few of the selected industry standard security protocols are discussed.

3.1 SAML2 Protocol

SAML2 provides a Web Browser SSO Profile of SAML2 which provides two bindings to handle the SSO. The HTTP POST Binding uses HTTP POST to send SAML token back to the Service/Resource Provider from Authorization Server (Identity Provider). The HTTP Redirect binding uses URL query string. SAML2 uses security tokens described in XML format for making assertions about the authenticated user.

3.1.1 Flow Diagram for Applications



3.1.2 Advantages and Limitations

1. HTTP Redirect may not be sufficient for bigger message assertions
2. SAML's HTTP POST Binding may not work well for native mobile applications
3. SAML works better for two-factor authentication use cases

3.1.3 Implementations

- Ping Identity
- Microsoft's Active Directory Federation Services (ADFS)
- SalesForce
- Apigee

- IBM Tivoli Access Manager
- Internet2 Shibboleth
- JBoss Federated SSO
- JOSSO
- Internet2 OpenSAML
- Entrust GetAccess
- Netegrity SiteMinder
- Oblix Netpoint
- RSA Security ClearTrust
- SunONE Identity Server
- Google Identity Platform
- AWS Directory Service

3.2 OAuth 2.0 Protocol

OAuth 2.0 is primarily an Authorization framework used to provide Resource access to Clients on behalf of Resource Owners. An Authorization Server is responsible for authenticating Resource Owners and authorizing resource access to Clients while the Resource Server is responsible for providing resources to Clients. The framework provides a process for end users to authorize third-party access to their resources without sharing their credentials using user-agent redirections. OAuth 2.0 define in RFC 6749 is a replacement of OAuth1.0 protocol and it is backward compatible with OAuth 1.0. The OAuth 2.0 framework is used for delegated user authorization and for building SSO solutions (through Open ID Connect protocol) for Enterprises. Security Access Tokens are used to provide Authentication and they can optionally have scopes that help define the access-level to Clients and/or EndUser on the Authorization server.

Clients go through a registration process with Authorization Server through a separate process. There are two types of Clients:

- Confidential Clients: Applications that can hold the state in a secure manner like Web Servers and use User-Agents (typically Web Browsers) for User Interface.
- Public Clients: They are primarily user-agent based applications like Single Page Apps, Silverlight, flash applications, Native mobile Applications

The framework provides options for four types of authorization grants:

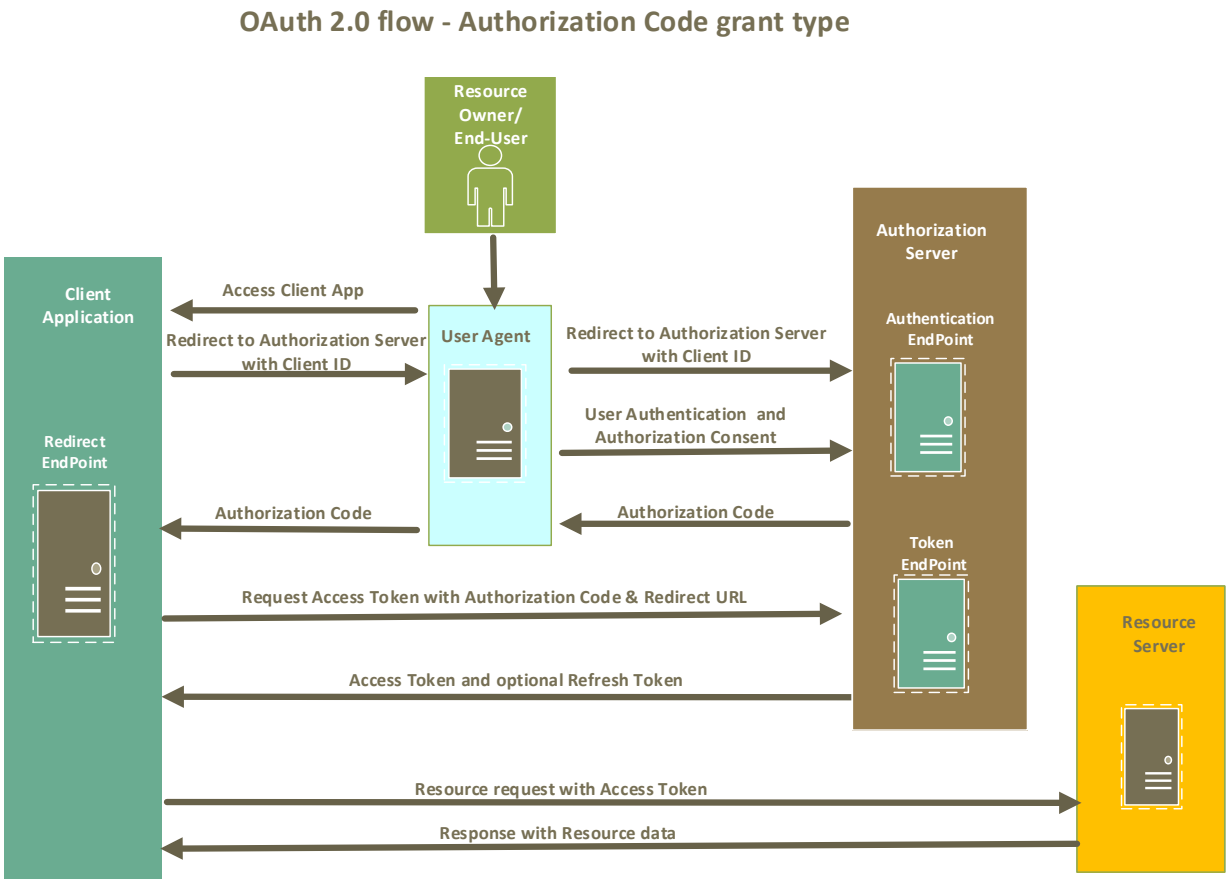
1. Authorization code: This grant type is primarily used compared to other grant types. It is used for Confidential Clients to obtain both Access tokens and Refresh tokens. To use this grant type, Clients need the ability to work with User-Agents (typically Web Browsers) and have the ability to receive callbacks. An Authorization Code represents the intermediate result of a successful Resource Owner/end-user Authorization process and is used by the Client to obtain Access/Refresh tokens. Authorization Codes are issued instead of tokens to the Clients as the code goes through an URL redirection involving the User-Agent.

2. Implicit Grant: The implicit grant type is used for Public Clients to obtain access tokens and is optimized for public clients known to operate a pre-registered redirection URI. This grant type does not support refresh tokens.

3. Resource Owner Password Credentials Grant: In this mode, the Client sends the Resource Owner's credentials (Username/Password) to the Authorization Server and is suitable for Clients (like device operating system) that are trusted by the Resource Owner. This is suggested to be used only when other grant types are not viable.

4. Client Credentials Grant: In this mode, the Client requests an access token using only its credentials when requesting access to protected resources under its control, or those of another resource owner that have been previously arranged with the authorization server.

3.2.1 Flow diagram using OAuth 2.0 Authorization Code grant type



3.2.2 Advantages and Limitations

- Requires using SSL/TLS for all connections.
- Provides ability to expose REST based Web APIs to Clients by Resource/Service Providers.

- Provides support for delegated User Authentication and as a base framework for Single Sign-On Solutions through Open-ID Connect.

3.2.3 Implementations

- Thinktecture Identity Server
- Apigee
- Ping Identity
- Microsoft's Azure Active Directory (Azure AD)
- SalesForce
- Google Identity Platform
- AWS Directory Service

3.3 OpenID Connect (OIDC)

OpenID Connect (OIDC) is an authentication layer built on top of OAuth 2.0 using TLS/SSL protocols. It is used by Clients/Relying Parties to authenticate end users by an Authorization Server or OpenID provider. In addition to authentication, Clients can obtain basic profile information in an interoperable and REST-based manner. OIDC internally uses JSON-based message flows with JSON Web Tokens (JWT) for digitally signatures. JWT is a compact, URL-safe means of representing end user claims to be transferred between two parties.

The framework provides options for three types of authentication flows:

1. Authorization code flow: This flow is used to obtain the Authorization Code that can be exchanged for access tokens and ID token. This is suitable for Clients that can maintain a client secret between themselves and the Authorization Server.

2. Implicit flow: The implicit flow is used for Clients, implemented in for a browser (User-Agent) written in JavaScript (like AngularJS), to obtain access tokens and ID tokens directly from the Authorization Server.

3. Hybrid flow: In this mode, some tokens are returned from the Authorization endpoint and some from the Token endpoint.

OpenID Connect is used for:

1. Federation of authentication for Enterprise applications to a centralized Identity Provider.

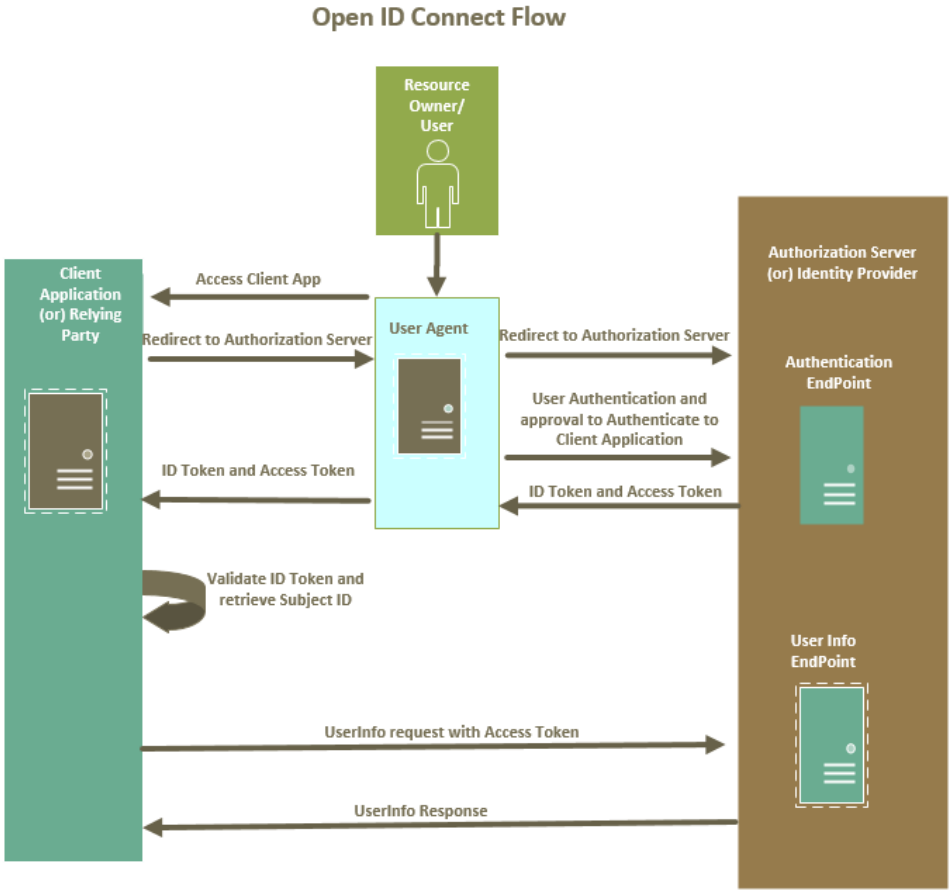
2. Providing a Single Sign-On (SSO) solution for enterprises by centralized authentication using Identity Providers.

3. Ability for an Enterprise to use third-party Identity Providers for Identity management that will help to:

    a. Reduce risk for Enterprises from loss of Identity Information from data breaches
    b. Use existing accounts of End users with the third-party vendors

c. Reduce costs and efforts of investing in Identity Management infrastructure.
d. Easily incorporate third-party SaaS applications into authenticated Enterprise applications for its end users
e. Allow enterprise applications to focus on focus on functionality and avoid baking identity management into applications.
f. Limit changes to enterprise applications as new advanced Authentication methods (like two-factor authentication) are developed.

3.3.1 Flow diagram for Open ID Connect



3.3.2 Advantages and Limitations

- Requires using SSL for all connections.
- Supports Browser based native apps and mobile applications for SSO solutions
- Used to expose REST based protected Web APIs to Clients by Resource and Service Providers.

3.3.3 Implementations

- Thinktecture Identity Server
- Apache mod_auth_openidc
- Google OAuth Client Library
- Openid-connect

- Drupal OpenId connect
- Shield
- Django OIDC
- Microsoft Azure Active Directory
- PingFederate from Ping Identity
- Amazon Web Services
- WSO2 Identity Server

4 Use cases

4.1 IAM for Enterprise level applications

4.1.1 Scenario

A Medical organization with physicians as users developed a Web Portal and a number of Web applications that are used for completion of certification activities as part of a Maintenance of Certification program or for Continuing Education. All the Web applications require user authentication for access. The organization wants users to log into the Web Portal and use it as an entry point to direct users to its other applications without the need for user to re-login into each of the applications.

4.1.2 Recommendations

a. The organization needs to adopt a centralized Identity management solution either self-hosted or through the use an off-site third-party Identity provider.
b. Enterprise Web Applications will not handle authentication at the application level explicitly but trust the Enterprise Identity provider issued security tokens/assertions/claims of the user to provide access.
c. The organization should not try to implement security protocols like OpenID Connect, OAuth2 or SAML2 on their own for a number of reasons: Effort, Efficiency, Accuracy, Maintenance and avoiding potential security breaches.
d. For self-hosting SSO Identity Providers
    - The organization can choose a commercial or open-source product for their SSO solution. Although both OpenID Connect and SAML2.0 can be used to provide SSO solution, our recommendation is to use OpenID Connect (built on top of OAuth2.0 framework).
    - This option allows organization to register their Applications with their own Identity provider.
    - User authentication/authorization is managed by a single Identity provider for the Enterprise.
e. For third-party hosted Identity Providers
    - The organization will not have any Identity information in their control.
    - The organization should prefer to choose cloud based SaaS Identity Providers for maximum flexibility and extensibility.
    - Choosing this option will relive the organization from IAM data concerns.

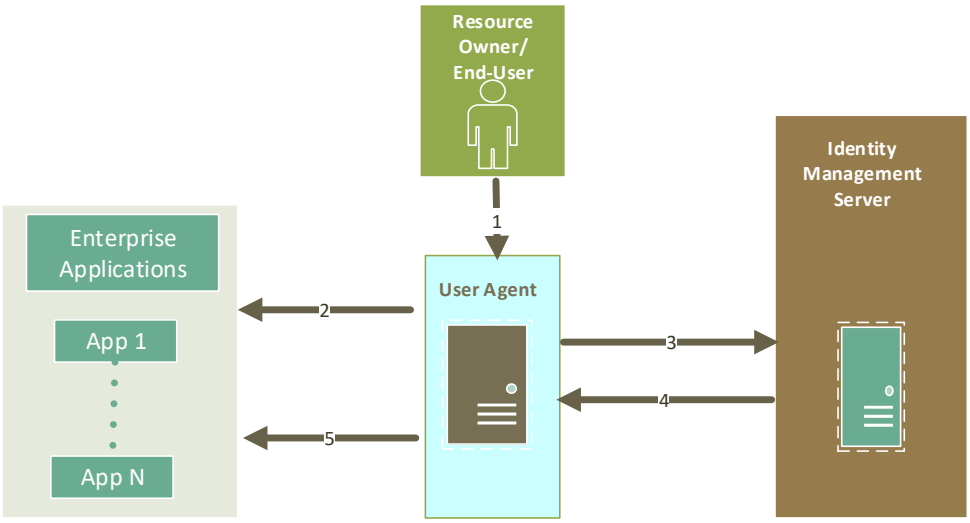
- This choice will allow the organization to enhance support for multiple third-party Identity Providers and in this case, the user will be provided with an option to choose one of the Identity Providers for authentication.
- Web applications are set to trust credentials from the configured Identity Provider(s).
- Our recommendation is to use OpenID Connect for an organization to federate Identity management.

f. If the organization, as part of its portfolio of applications, develops native Mobile applications and/or browser based Single Page Apps (SPA)

- The organization should choose Identity providers that implements OpenID Connect as the Mobile/ SPA applications will not be able to leverage SAML2.
- The organization can take advantage of existing OpenID Connect client libraries available for use in native Mobile or SPA applications.

4.1.3 End-User interaction diagram

High level user interaction flow diagram of the recommended solution is shown below



4.2 IAM for applications between Enterprises

4.2.1 Scenario

A medical organization A developed a Web Portal and number of Web applications for their customer base of Physicians for completion of certification activities as part of Maintenance of Certification program. Another organization B developed a certification activity hosted as a web application C. Organization A has an agreement with Organization B to allow their physicians to complete the activity using Web Application C and get credit. Organization A wants their users to access Organization B's activity using the same login as they use for the applications of Organization A.

4.2.2 Recommendations

a. Organizations A and B need to provide support for Federated SSO with a trust relationship as each organization maintains their user's identity information in their respective Identity providers.

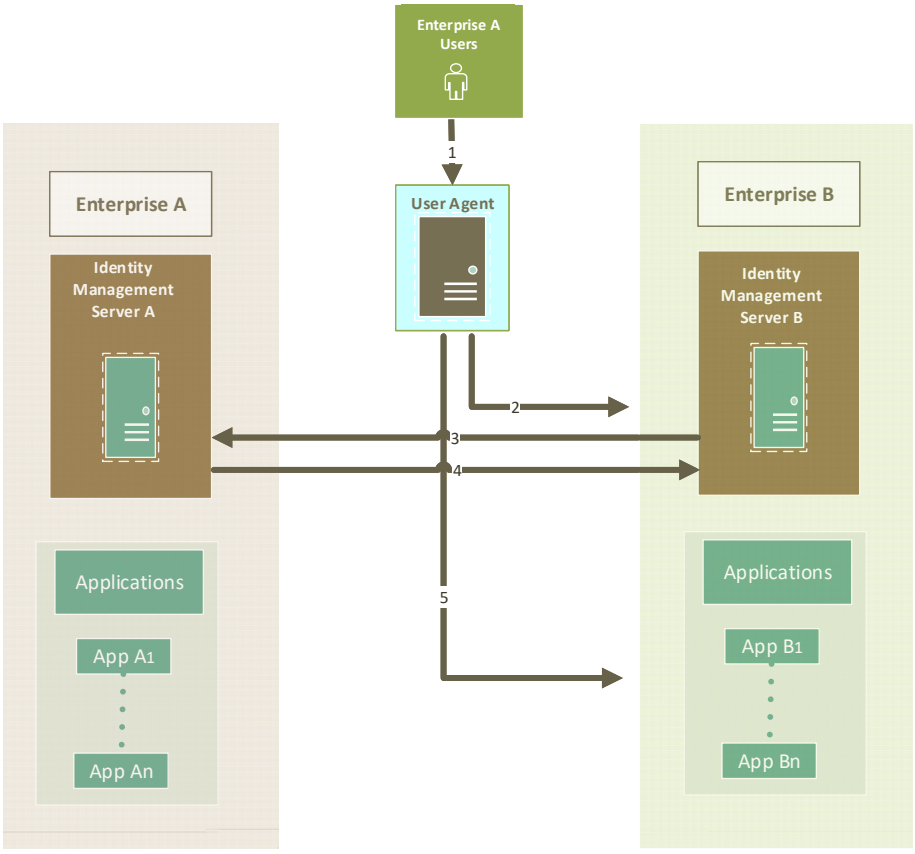
b. High level flow:
   - Users of Organization A will be redirected to their regular authentication page when they access Web Application C of Organization B.
   - Once the user is logged in, the user's identity information flows to Organization B's Identity Provider which verifies that the user claims are from Identity provider of Organization A and following that will translate the user claims that are expected by application C.
   - Application C verifies the user's claims and grants access to the application based on the provided claims.
   - Authenticated Organization A users will be able to access the Web application of Organization B seamlessly.

c. Our recommendation is to use OpenID Connect (OIDC) for Federated SSO to support in this scenario.

d. Organizations A and B can choose a commercial or open-source product for their Federated SSO solution that implement one of OpenID Connect or OAuth2 or SAML2 if trust can be established between the Identity Providers of organizations A and B.

4.2.3 End-User interaction diagram

High level user interaction flow diagram of the recommended solution is shown below



4.3 IAM for Enterprise resource access provided through Web API

4.3.1 Basic Scenario

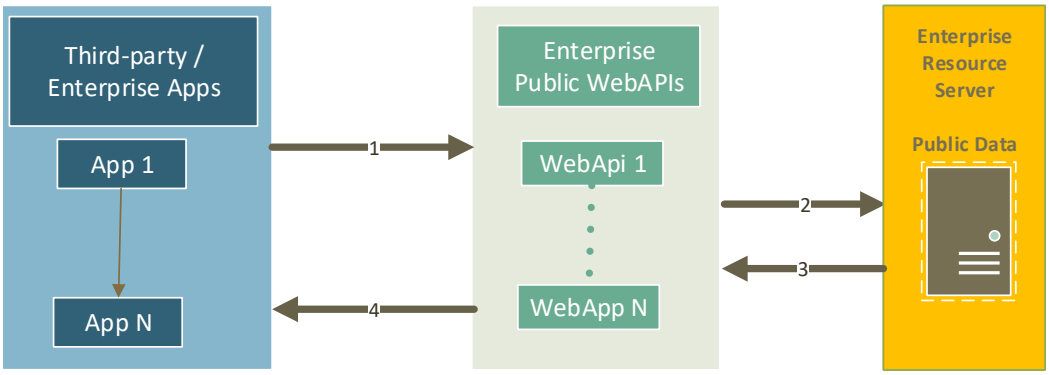
A Medical organization is the primary source of reference data for information such as Medical Schools or Residency Programs and wants to expose this data to allow third-party applications to consume the reference data.

4.3.2 Basic Scenario Recommendation

a. Organization needs to develop a WebAPI with appropriate end-points to provide the reference information in its entirety or by business key with additional support for filters
b. The WebAPI needs to be hosted using transport level security (SSL/TLS) to provide data privacy, integrity and non-repudiation.
c. The MedBiquitous API architecture should be followed for the design of the services.

4.3.3 Application flow diagram

High level application flow diagram of the recommended solution is shown below



4.3.4 Extended Scenario – User Delegation

A Medical organization with physicians as users develop a Web Portal and number of Web applications that are used for completion of certification activities as part of a Maintenance of Certification program or for Continuing Education. The organization wants to allow third-party applications to consume a physician's Certification Verification and Certification History data upon End-User consent.

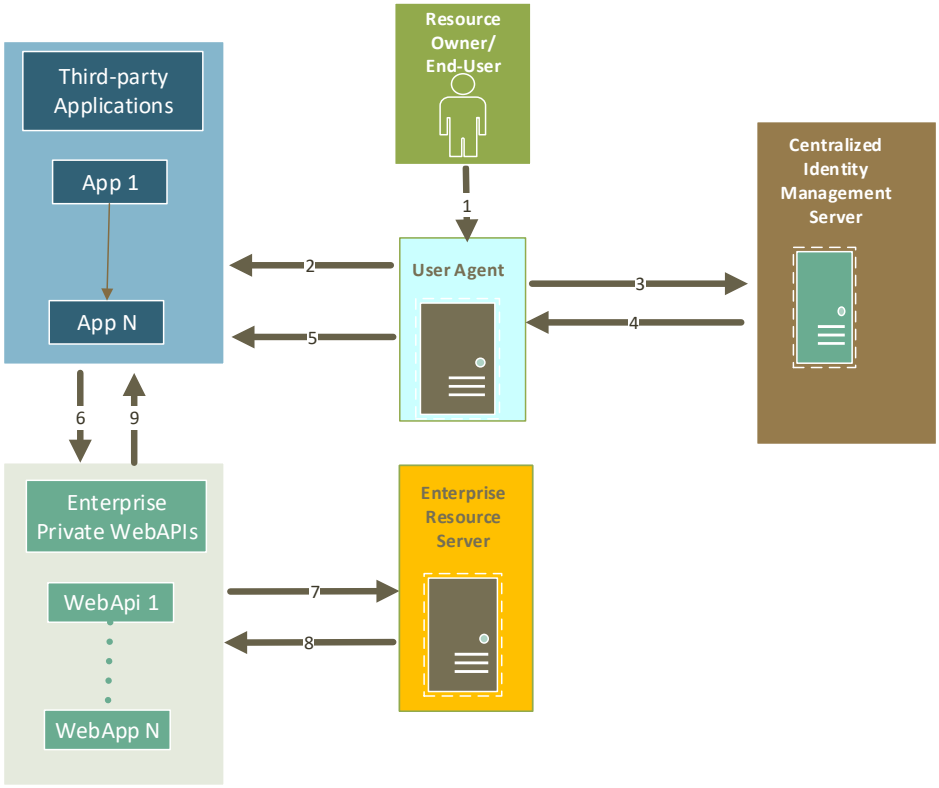
4.3.5 Extended Scenario Recommendation

a. Organization needs to adopt a centralized Identity Management solution as described in 4.1. However, the Identity Provider needs to support OAuth2 workflow to handle authentication and authorization grants.
b. The organization creates a Web API to expose the user's data to the third-party applications for consumption.
c. The WebAPI requires authentication of the organization's user and authorization from the user.
d. Access to user's data is provided to the third-party application based on user consent.
e. The WebAPI will not handle authentication at the application level but trust the organization's Identity Provider-issued security claims.

f. Third-party applications follow the OAuth2.0 Authorization workflow where the User is authenticated by the organization's Identity Provider and then presented with a request for consent to authorize the third-party application to access the user's data.
g. This option will allow organizations to grant access to their user's data to any third-party applications that support the OAuth2.0.
h. If the organization chooses third-party hosted Identity Providers or cloud based SaaS identity Providers, it will provide maximum flexibility for integration with third-party applications.
i. The MedBiquitous API architecture should be followed for the design of the services.

4.3.6 Extended scenario End-User interaction diagram

High level user interaction flow diagram of the recommended solution is shown below



4.4 IAM for integrating third-party applications with Enterprise applications

4.4.1 Scenario

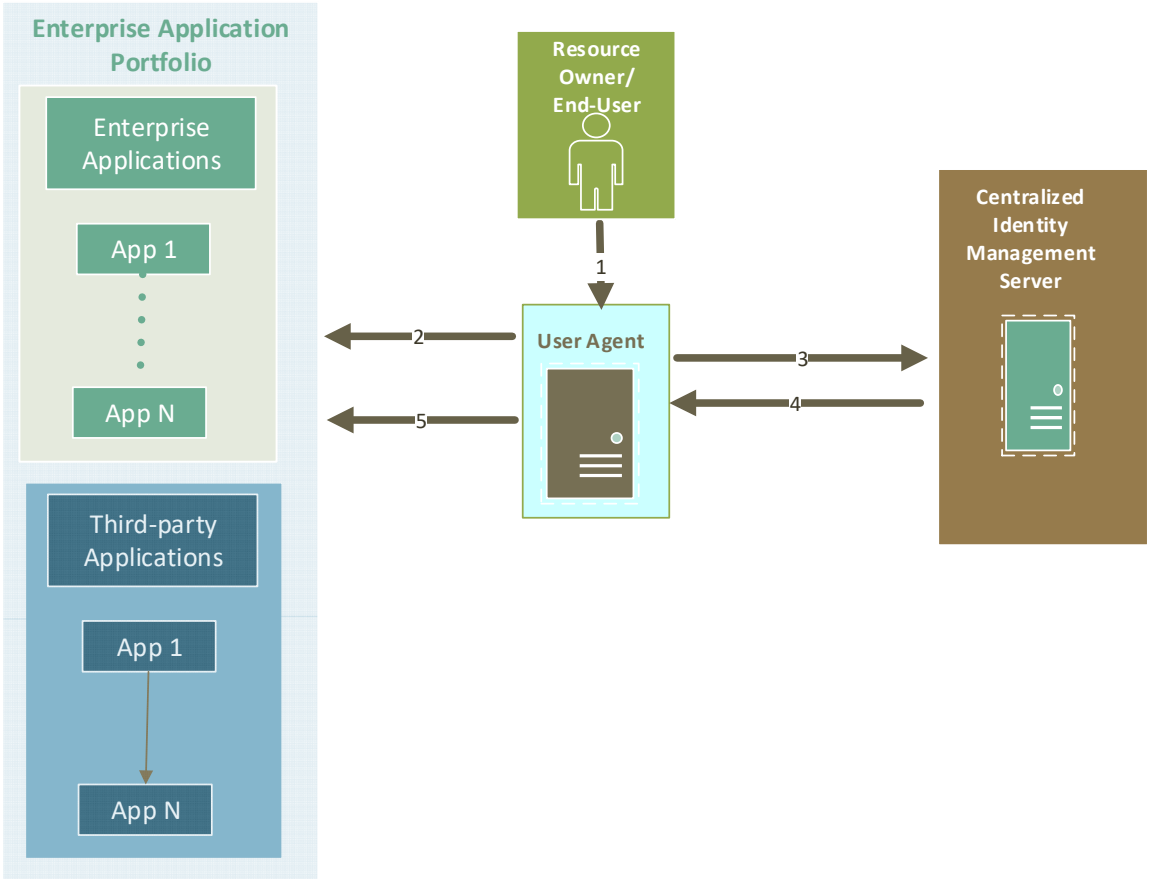
A Medical organization with physicians as users developed a Web Portal and number of web applications that are used for completion of certification activities as part of Maintenance of Certification program or for Continuing Education. The organization wants to enhance its portfolio of applications available to the user to meet their requirements by approving third-party applications. The organization wants their users to seamlessly access third-party applications once they are authenticated in the organization's Web Portal without the need to re-login.

4.4.2 Recommendation

a. The organization needs to adopt a centralized Identity management solution as described in 4.1.
b. Third-party applications require Authentication but they do not implement the authentication process and instead rely on the organization's Identity Provider(s) for authentication.
c. If the third-party applications approved by the organization as part of its portfolio of applications develops native Mobile applications and/or browser based Single Page Apps (SPA).
   - The organization must choose Identity providers that implement OpenID Connect protocol.
   - The organization can take advantage of existing OpenID Connect client libraries available for use in the native Mobile or SPA applications

4.4.3 End-User interaction flow diagram

High level user interaction flow diagram of the recommended solution is shown below



5 Industry Traction and future direction

The need for enterprises and businesses to provide a Single Sign-On (SSO) solution to allow a single login from a user to grant access to all their applications has increased in the last decade. With the digital revolution and connectivity, there is a greater need for organizations to support a Federated SSO that enables smooth integration with external partners and to incorporate third-party and/or commercial products into their application portfolio. Organizations have valid business cases to

expose data through WebAPI's to external partners and third-party applications with Federated SSO playing a pivotal role in addressing the security aspect of this objective. Open ID Connect protocol is gaining greater momentum for providing SSO and Federated SSO solutions. OAuth2.0 protocol is gaining momentum to support delegated user authorization use cases for Enterprises. SAML2.0 protocol released initially in 2005, while more powerful but is not focused on non-browser based applications and browser based Single Page Apps that are getting more prevalent in industry. To circumvent these limitations of SAML2.0, some cloud providers are incorporating SAML2.0 as part of OAuth2.0 workflow for performing authentication. SAML2.0 does not have same traction and adoption rate as OAuth2.0 and Open ID Connect. As organizations realize the effort needed to keep up with identity information security threats, they may gravitate towards replacing their internal Identity Providers with cloud based external Identity Providers to completely out-source identity management.

6 References:

- OAuth2.0
- OAuth 2.0 IETF - RFC 6749, RFC 6750 and RFC 6819
- OAuth 1.0 protocol
- OpenID Connect
- WSO2 – OpenID Connect
- OIDC Simple Client Profile
- OASIS SAML2 profiles
- Gatner survey on SSO
- Cloud SSO
- Open ID Connect Libraries
- Shibboleth
- JOSSO
- JBoss Federated SSO
- Microsoft Azure Active Directory
- Google APIs OAuth2.0
- SalesForce SSO
- Article on SSO strategy
- Article on Federated Identity Management
- Article on SAML and OAuth2
- Article on SSO Strategy
- Article on Traditional SSO vs Federated SSO
- Article on Federated Identity Management