UMA Profile for OpenID Connect Specification

# Introduction

UMA Claims 2.0 approach is designed to support claims-based Access Control. In other terms, the access control decision to grant access to Authorizing User's resource (Protected Resource at Host) is made based on Requesting party information, such as Subject's name, age (or date of birth) email address, role, location, or score credit, etc.

## Purpose

This specification defines UMA interface to interact with OpenID Connect protocol in order to support claim-based access control.

## Scope

* Benefits, objectives and goals

## Definitions

Authorization Manager

Requesting Party

Authorizing User

Authorization Endpoint

Claims

Protected Resource

Host

OpenID

## References

User-Access Manager Core specification

OpenID Connect Protocol specification

## Overview

The primary driver for UMA Claims 2.0 is the process of negotiation for access authorization defined by the User-Managed Access (UMA) core protocol, in which an authorization manager can require a requester to convey claims on behalf of a requesting party, in order to satisfy the polices of an authorizing user.

OpenID Connect provides authentication, authorization, and attribute transmission capability. It allows third party attested claims from distributed sources.

The following diagram shows the conceptual model about the integration scenario between UMA and OpenID Connect protocols.

This approach uses OpenID Connect protocol to request Requesting Party’s claims, leveraging specific mechanisms to transmit attributes from distributed sources.

As showed in the diagram, the concept includes two domains:

* UMA Domain where the Authorizing User protects (using an Authorization Manager) its own web resources (Host).
* OpenID Domain which provides authentication and SSO mechanisms for the Requesting Party , including authorization services to allow access to third party UserInfo EndPoint (Protected Resource).



UMA Authorization Manager (AM) interacts with OpenID Domain to request claims based on the OpenID Connect specification, which includes the following high level steps:

1. The Client sends a request to the Server’s End-User Authorization Endpoint.
2. The Server authenticates the user and obtains appropriate authorization.
3. The Server responds with access\_token and a few other variables.
4. The Client sends a request with access\_token to the Userinfo Endpoint.
5. Userinfo Endpoint returns the additional user supported by the Server.

# Overall description

The following diagram shows the overall architecture about the UMA AM Claim 2.0 and OpenID Connect protocol integration strategy.

UMA AM provides two specific interfaces:

* OpenID 2.0 Relying Party (RP).
* Claims Client for OpenID Connect

OpenID 2.0 Relying Party interface is responsible to authenticate the Requesting Party and initialize the OpenID Connect protocol.

Claims Client interface is responsible to request claims based on OpenID Connect protocol, in order to satisfy claim-based policy. The Client interacts with the Authoriation Server to obtain specific access token to access to the Requesting Party’s UserInfo EndPoint.

OpenID Identity provider provides an OpenID 2.0 authentication infrastructure and an Authorization Server, as described in the OpenID Connect specification, in order to handle Claims request form the UMA AM Claims Client.

UserInfo EndPoint that is a protected resource by the Authorization Server (OpenID Connect protocol), provides user attributes (claims).

The Requesting Party interacts with UMA AM to convey claims and with OpenID Identity Provider/Authorization Server to authorize claims release from the UserInfo EndPoint.



The following steps describe the high-level claim-based access control, based on previous architecture:

Assumptions:

* Alice (Authorizing User) has set a claims-based policy at AM to restrict access to a protected web resource.

Goal:

* Alice wants permit the access to the resource to Bob if and only if Bob has an email account equal to “bob@gmail.com”.

Steps:

1. Requesting Party (Bob) attempts to access to a protect resource at Host.
2. Host redirects Bob to the Authorization Manager (UMA AM) to apply claim-based policy.
3. Bob is authenticated through OpenID mechanism at AM site (Bob interacts with the OpenID provider).
4. AM (Claims Client) sends a request to the Server’s Requesting Party (End-user) Authorization Endpoint at the OpenID Provider.
5. The Server authenticates the Requesting Party (the user has a SSO session) and obtains appropriate authorization.
6. The Server responds with access\_token and a few other variables.
7. AM sends a request with access\_token to the Userinfo Endpoint.
8. Userinfo Endpoint returns the claims supported by the Server.
9. UMA AM verifies the returned claims if they match with the policy.

## Claims Request

TBD

## Claims Response

TBD