Extending UMA Protocol to support Trusted Claims (tClaims)

Newcastle University

Domenico Catalano
and the Smart Team
Agenda

- UMA Conceptual model
- tClaims Requirements Analysis
- OpenID Connect
- UMA/OpenID Connect Integration approach
- User Interaction
- Trust Model consideration
- Q&A
UMA Conceptual Model

- **UMA AM**
  - policy decision point

- **Host**
  - Manage

- **Authorizing User**

- **UMA Domain**

- **Requester**
  - Protect
  - Authorize

- **Requesting Party**

- **External Domain**

- **Protected Resource**
  - Access
UMA Trusted Claims

- UMA Trusted Claims approach is designed to support Claims-based Access Control.

- In a Claim-based Access Control, the decision to grant access to a protected resource is made based on Subject’s information, such as name, age, email address, role, location, or credit score, etc.
tClaims example scenarios

- Enterprise class scenario
  - Accessing Personal Loan Special Program

- Social/web class scenario
  - Sharing photo with “bob@gmail.com”
Accessing Personal Loan special program

Enterprise Class Scenario

• Bank online service provides an User-Managed Claims access control to restrict and personalize access to special program/service (i.e. personal loan with low interest rate) to users which have determinate employment (i.e. government employee), and have an high credit score.
Alice at Bank site for requesting access to a restricted service. An UMA protected resource.
Sharing Photo with “bob@gmail.com”

Social/web Class Scenario

- Alice wants share a photo gallery with bob if Bob has an account email “bob@gmail.com” and he is 18 years old.
Alice defines claims-based authorization policy, using In-App widget.
Requirements Analysis

- Authorizing User (Resource Owner) needs a claims-based access control to restrict access to own resources based on Requesting Party’s Identity attributes.

- Identity attributes must be issued by a Trusted Third Party (TTP) and verifiable by a Claims Requester.

- Claims may be logically aggregated to provide a collection of attributes from different Attribute Providers (Claims Host).
OpenID Connect

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

- This specification is largely compliant with OAuth 2.0 draft 15.
OpenID Connect protocol overview

- OpenID Connect protocol in abstract follows the following 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.
UMA Conceptual Model

- **UMA AM** (policy decision point)
- **Host**
- **Requestor**
- **Requesting Party**
- **External Domain**
- **UMA Domain**
- **Authorizing User**

**Paths:**
- Control
- Protect
- Manage
- Authorize
- Protected Resource
- Access
UMA Conceptual Model with tClaims

1. Request
2. AuthN AuthZ
3. Access_token
4. Request Userinfo
5. Userinfo

UMA AM
- Claims
- Control
- Protect
- Manage

OpenID Connect AS
- UserInfo
- Protect
- SSO

Authorizing User
- UMA Domain
- Policy decision Point
- Protect

Requesting Party
- OpenID Domain
- Requester
- Request

HOST
- Requester
- Protected Resource
- Access

Client

UMA Domain

OpenID Domain
UMA/OpenID Connect Integration approach
User eXperience
Scenario

Sharing Photo with “bob@gmail.com”

- Host In-App Fast Sharing settings.
- Requesting Party requests direct access to Protected Resource.
- OpenID Connect interaction.
Alice at Host Site

Protected Resource by CopMonkey AM

in-App Fast AuthZ Settings for sharing
Alice defines claims-based authorization policy, using In-App widget
Protected Resource is ready for sharing under authZ policy
Alice shares the Protected resource through twitter
Bob attempts to access to protected resource.

Bob is redirect to AM to convey claims.
CopMonkey authenticates Bob through OpenID, in order to initialize OpenID Connect protocol
Bob is redirect to IdP’s authorization service to grant claims.
Bob gets access to the protected resource
Trust Model Consideration
Bootstrapping Trust

A
Self-Registration

B
Affiliate or SSO

C
Affiliate or SSO with Trusted Framework
Thanks