Seeking Sponsors: UMA Project

• UMA = User Managed Access
  – Extension or profiling of OpenID-Connect & OAuth2.0

• UMA is a Working Group in Kantara Initiative
  – http://kantarainitiative.org/confluence/display/uma
  – http://kantarainitiative.org/confluence/display/uma/UMA+1.0+Core+Protocol

• Specs contributed to IETF OAuth2.0 WG:
  – http://tools.ietf.org/id/draft-hardjono-oauth-umacore

• Binding Obligations specification:
  – Identifies legal obligations of players within the protocol flow
  – http://docs.kantarainitiative.org/uma/draft-uma-trust.html

• Implementations available (incomplete)
  – Gluu, CloudIdentity (UK)
UMA: User-Centric Resource Sharing
UMA: Single Point for Managing Access

Alice uses the UMA Authorization Server to set access-policy for her resources.

Bob uses the Client to get authorization from the AS, to access Alice’s resources.
UMA Phase-1: Alice registers Resource

(1) Alice registers her Resource (e.g. files) Stored at RS#1 to the AS & sets access Policy

(2) & (3) The Resource Server #1 obtains an OAuth2.0 PAT Token from the Authorization Server.
UMA Phase-2: Client gets authorization token

1. Clients attempts access without token
2. Client redirected to AS.
3. AS issues AAT token to Client
UMA Ph-3: Requesting Party gets Authorization

(1) & (2) Bob attempts access without token
(2) Client redirected to AS.
(3) AS issues RPT token to Bob (the human)
(4) Authorization Server

Alice

Resource Owner

Bob

Requesting Party

Client (Requester)

Authorization Server

Alice’s Account

Protected Resource (eg. files)

Resource Server #1

Protected Resource (eg. files)

Resource Server #2

Protected Resource (eg. files)

Resource Server #3
UMA: Bob successfully accesses Alice’s files

(1) & (2) Bob uses Client to access files, presenting RPT and AAT token
UMA: Relationship with OAuth2.0 & OIDC

**OpenID Connect**
- You achieve federated single sign-on and login-time attribute exchange
- You control access to claims about you

**OAuth 2.0**
- You control access to web APIs
- Apps can use a variety of access token types

**UMA**
- You can grant access to apps operated by anyone
- You control access to a variety of protected resources
- You can grant access by setting policies and terms ahead of time
- The authorization function is standard and centralizable
- Requesting party is authorized based on claims

References normatively as an option (sharing some features as a result)

Source: Eve Maler & UMA WG