User-Managed Access (UMA)

http://tinyurl.com/umawg
@UMAWG
10 December 2010
Privacy is not about secrecy

“The goal of a flexible, user-centric identity management infrastructure must be to allow the user to quickly determine what information will be revealed to which parties and for what purposes, how trustworthy those parties are and how they will handle the information, and what the consequences of sharing their information will be.”

– Ann Cavoukian, Information and Privacy Commissioner of Ontario, *Privacy in the Clouds* paper

It’s about context, control, choice, and respect
Digital identity management

Vendor relationship management

Online social networking

differentiated app behavior based on permissioned data sharing

digital shadow cruft
self-determination

privacy

informational
determination

user
centricity

differentiated
app behavior
based on
permissioned
data sharing

digital identity
management

digital shadow cruft

vendor relationship
management

online social
networking

policy decision-making
digital identity management

vendor relationship management

differentiated app behavior based on permissioned data sharing

online social networking

data portability

the "Open Stack"

the "Connect" phenomenon

digital shadow cruft

policy decision-making
privacy
informational self-determination
user centricity
personal datastores

volunteered personal information

vendor relationship management

differentiated app behavior based on permissioned data sharing

online social networking

data portability

the "Open Stack"

the "Connect" phenomenon

digital identity management

user centricity

informational self-determination

privacy

policy decision-making
UMA is...

- A web protocol that lets you control authorization of data sharing and service access made on your behalf
- A Work Group of the Kantara Initiative that is free for anyone to join and contribute to
- A set of draft specifications that is free for anyone to implement
- Undergoing multiple implementation efforts
- Slated to be contributed to the IETF
- Striving to be simple, OAuth-based, identifier-agnostic, RESTful, modular, generative, and developed rapidly
<table>
<thead>
<tr>
<th><strong>Selective sharing shortcomings</strong></th>
<th><strong>User-Managed Access solutions</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Little sophistication and consistency in Web 2.0 access control – e.g., Google Calendar vs. Flickr vs.TripIt</td>
<td>A way for any web app to provide sophisticated access control merely by outsourcing it, à la SSO</td>
</tr>
<tr>
<td>Rules for selective sharing can’t be applied to different apps – the “family” ACL has to keep being rebuilt</td>
<td>Selective-sharing policies can be mapped to content at multiple hosts</td>
</tr>
<tr>
<td>Selective sharing is largely identity-based and static</td>
<td>Conditions for access can be “claims-based”, with claims tested when access is attempted – e.g., “anyone over 18”</td>
</tr>
<tr>
<td>Individual is only in a position to consent to sharing and to site terms of service, not dictate terms of access</td>
<td>Sharing policies form a barrier; requesting parties have to agree to terms or otherwise prove suitability</td>
</tr>
<tr>
<td>Individual can’t get a global view of every party they’ve said can get access to their data and content</td>
<td>Sharing policies and sharing authorizations all come out of a single “hub” application</td>
</tr>
<tr>
<td>OAuth today only enables the protection of singular API endpoints for web services</td>
<td>Any Web resource with a URL, and any access scope on it, can be protected – e.g., sharing a status update API or a single tweet</td>
</tr>
</tbody>
</table>
UMA players
(see also UMA Explained wiki page)
UMA players
(see also UMA Explained wiki page)

A web user who configures an authorization manager with policies that control how it makes access decisions when a requester attempts to access a protected resource at a host.
UMA players
(see also UMA Explained wiki page)

a web user who configures an authorization manager with policies that control how it makes access decisions when a requester attempts to access a protected resource at a host

enforces access to the protected resources it hosts, as decided by an authorization manager
UMA players
(see also UMA Explained wiki page)

- **Authorizing User**: A web user who configures an authorization manager with policies that control how it makes access decisions when a requester attempts to access a protected resource at a host.

- **Host**: Carries out an authorizing user's policies governing access to a protected resource.

- **Authorization Manager**: Manages, controls, and authorizes access to the protected resources it hosts, as decided by an authorization manager.

- **Requester**: Enforces access to the protected resources it hosts, as decided by an authorization manager.

- **PEP**: Protects the Protected Resource.

- **PDP**: Determines whether the access is authorized.

- **Requesting Party**: Delegates the management and control of access decisions to the Authorization Manager.
UMA players
(see also UMA Explained wiki page)

a web user who configures an authorization manager with policies that control how it makes access decisions when a requester attempts to access a protected resource at a host
carries out an authorizing user's policies governing access to a protected resource

a web user, or a corporation or other legal person, that uses a requester to seek access to a protected resource

enforces access to the protected resources it hosts, as decided by an authorization manager
UMA players
(see also UMA Explained wiki page)

- **Authorizing User**: A web user who configures an authorization manager with policies that control how it makes access decisions when a requester attempts to access a protected resource at a host.

- **Host**: Enforces access to the protected resources it hosts, as decided by an authorization manager.

- **Authorization Manager**: Carries out an authorizing user’s policies governing access to a protected resource.

- **Requester**: Seeks access to a protected resource.

- **Requesting Party**: A web user, or a corporation or other legal person, that uses a requester to seek access to a protected resource.
UMA has three steps

1. Trust a token
   • Alice introduces her Calendar host to CopMonkey: “When CopMonkey says whether to let someone in, do what he says”

2. Get a token
   • A travel marketing company tries to subscribe to Alice’s calendar but it has to agree to her terms of use: “All right, all right, I’m clicking the ‘I Agree’ button”

3. Use a token
   • The marketing company now has an OAuth access token to use at the Calendar host: “This means Alice thinks it’s okay”
The players again

Authorizing User (user at browser or other user agent)
Step 1 protocol flow

Step 1. User Introduces Host to AM

Authorizing User (user at browser or other user agent)
A possible UX for host-AM introduction
Step 2 protocol flow

**Step 1. User Introduces Host to AM**

**Step 2. Requester Gets Access Token**

**Authorizing User** (user at browser or other user agent)
A possible UX for self-asserted claims about promises

- You must acknowledge to be over 18 years old to be granted access to this resource.
- You must acknowledge to adhere the Creative Commons licensing terms to be granted access to this resource.

Confirm
A potential claims trust model: make them UMA-protected resources
Step 3 protocol flow

1a. provision AM location

1b. get metadata

1c. authorize Host to trust AM

1d. register resources

1e. define policies

2a. attempt access

2b. ask for access token, supplying claims as demanded

3b. validate token

Ia. provision resource

Ib. get metadata

Ic. authorize Host to trust AM

Authorization Manager (AM)

Authorization Server

Protected Resource

metadata

OAuth 2.0

Client

Protected Resource

2a. provision Resource location

2b. provision Access Token

Requesting Party

Requester

Step 1. User Introduces Host to AM

Step 2. Requester Gets Access Token

Authorizing User (user at browser or other user agent)
Status of UMA development

(see also Working Drafts wiki page)

Work incubating in UMA WG

Simple Access Authz Claims

Claims 2.0

UMA Core

Resource Registration

OAuth 2.0

Dynamic discovery/registration

Token revocation

Signatures

Redelegation

trust claims

Legal considerations

Scenarios and use cases

Requirements

User stories

hostmeta

XRD .well-known

Work taking place in IETF
Status of UMA development

(see also Working Drafts wiki page)

- Work incubating in UMA WG
  - Simple Access Authz Claims
    - trusted claims
    - Claims 2.0
    - UMA Core
      - legal considerations
      - Resource Registration
        - user stories
      - scenarios and use cases
    - requirements
  - dynamic discovery/registration
  - XRD .well-known
  - OAuth 2.0
    - token revocation
    - signatures
    - redelegation
      - trusted claims

- Work taking place in IETF
  - OAuth 2.0
  - OAuth Core
  - OAuth 2.0 Authorization Framework
  - OAuth 2.0 Core
  - OAuth 2.0 Authorization Framework
  - OAuth 2.0 Core

- UMA Validator Bounty Program
Thanks!
Join us!

http://tinyurl.com/umawg
@UMAWG

Special thanks to Maciej Machulak and Domenico Catalano for screenshots and graphics