User-Managed Access (UMA) WG Update

Cigdem Sengul | @UMAWG | tinyurl.com/umawg

Q2 2019
OAuth enables constrained delegation of access to apps

Benefits:
• Flexible, clever API security framework
• Alice can agree to app connections and also revoke them
OpenID Connect does modern-day federation

Benefits:
- **Layers** identity/authentication tech with delegation/authorization tech
- **Translates** federated identity for mobile and the API economy

Authorization server

Resource server

Client

Federation user

Relying party

Identity provider (OP)

Standard UserInfo endpoint
To OAuth, UMA adds cross-party sharing...

Benefits:
- Secure delegation
- Alice can be absent when Bob attempts access
- Helpful error handling for client applications
...in a wide ecosystem...

Benefits:
- Alice **controls trust** between a service that hosts her resources and a service that authorizes access to them.
...of resource hosts

Benefits:
- Resource hosts can outsource authorization management – and liability – to a specialist service
- Alice can manage sharing at a centralizable service
- Bob can revoke his access to Alice’s resources
UMA user experience opportunities

- UMA Resource owner
- Ahead of time: Share
- Anytime: Monitor, Withdraw
- At run time: Opt in
- After the fact: Approve

Confidential App is requesting permission to access:
- Access and change your email contacts

Options: Allow Access, No thanks.
### Benefits for service providers

<table>
<thead>
<tr>
<th>True secure delegation; no password sharing</th>
<th>Scale permissioning through self-service</th>
<th>Resources accessed from distributed locations</th>
<th>Foster compliance through standards</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="True secure delegation; no password sharing" /></td>
<td><img src="image2.png" alt="Scale permissioning through self-service" /></td>
<td><img src="image3.png" alt="Resources accessed from distributed locations" /></td>
<td><img src="image4.png" alt="Foster compliance through standards" /></td>
</tr>
</tbody>
</table>

- True secure delegation; no password sharing
- Scale permissioning through self-service
- Resources accessed from distributed locations
- Foster compliance through standards
Benefits for individuals

- Choice in sharing with other parties
- Convenient sharing/approval with no outside influence
- Centralizable monitoring and management
- Control of who/what/how at a fine grain
Known implementations

(more detail at tinyurl.com/umawg)

• ForgeRock – financial, healthcare, IoT, G2C...
• Gluu (open source) – API protection, enterprise, G2C...
• ShareMedData – healthcare
• HIE of One / Trustee (open source) – healthcare
• IDENTOS – healthcare, G2C
• Pauldron (open source) – healthcare
• RedHat Keycloak (open source) – API protection, enterprise, IoT...
• WSO2 (open source) – enterprise...

Interop report session upcoming at Identiverse in June
Typical use cases

Profiles / references:
- Health Relationship Trust
- UK Pensions Dashboard
- OpenMedReady Alliance

Alice-to-Bob (person-to-person) delegated sharing of health data/devices, financial data, connected cars...

Enterprise-initiated delegated sharing – enterprise API access management, access delegation between employees

Alice-to-Alice (person-to-self) delegated sharing – proactive policy-based sharing of OAuth-style app connections
Relevance for privacy beyond individual empowerment

• Features relevant to privacy regulations:
  • Asynchronous control of access grants
  • Enabling Alice to monitor and manage grants from a “dashboard”
  • Auditability of grants (consent) and interactions between an AS and each RS

• Work is well along on an UMA business model
  • Modeling real-life data-sharing relationships and legal devices
  • Technical artifacts are mapped to devices
  • Goal: tear down artifacts and build up new ones in response to state changes
The UMA business model

- **Resource Rights Administrator**
  - **Delegates-perm-authority-to**
- **Authorization Server Operator**
  - **Delegates-perm-authority-to**
- **Resource Server Operator**
  - **Licenses-perm-granting-to**
  - **Licenses-perm-getting-to**
- **Requesting Agent**
  - **Licenses-perm-getting-to**
  - **Permits-knowing-claims**
  - **Delegates-seek-authority-to**
- **Data Subject**
  - **Delegates-perm-authority-to**
- **Client Operator**
  - **Licenses-perm-getting-to**
UMA in a nutshell

- Developed at **Kantara Initiative**
  - V1 done in 2015, V2 done in 2018

- Leverages existing **open standards**
  - OAuth2
  - OpenID Connect and SAML (optional but popular)

- Specs **contributed** to IETF OAuth WG in Feb

- Profiled by multiple **industry sectors**
  - Financial, healthcare, government

- UMA business model effort supports **legal licensing** for personal digital assets

- Some 1:1 **interop testing** done; more soon?
UMA implications…

...for the client
- Simpler next-step handling at every point

...for the RS
- Standardize management of protected resources
- Control data sharing/device control
- Truly delegate access to other parties using clients

...for the RO
- Offer interoperable authorization services
- Don’t have to touch data to protect it

...for the AS
- Truly delegate access to other parties using clients
- Offer interoperable authorization services
- Don’t have to touch data to protect it

...for the RqP
- Seek access to a protected resource as oneself

...for the client operator
- Distinguish identities of resource owners from mere users

...for the resource server operator
- Externalize authorization while still owning API/scopes

...for the resource rights admin
- Manage sharing on behalf of data subjects, not just for oneself

...for the authorization server operator
- Prove what interactions took place or didn’t
- Revoke access (or request it) to someone else’s assets

...for the requesting agent
- Revoke access (or request it) to someone else’s assets
Join us! Thank you! Questions?

@xmlgrrl | @UMAWG | tinyurl.com/umawg