GDPR, PSD2, CIAM, and the Role of User-Managed Access 2.0

Eve Maler
VP Innovation & Emerging Technology, ForgeRock
@xmlgrrl | eve.maler@forgerock.com
Chair and founder, Kantara UMA Work Group
@UMAWG | tinyurl.com/umawg
24 Jan 2018
The new business imperatives
LIKE A BOSS
We're all individuals!

I'm not
What makes data privacy regulations different this time around?

• Virality
• Digital transformation
• Aspirations
Take steps...
1. Find where digital transformation opportunities and user trust risks intersect
2. Conceive of personal data as a joint asset
3. Lean in to consent
4. Take advantage of identity and access management for building trust
UMA 101
OAuth is for constrained delegation to apps
It has helped to kill the “password anti-pattern”

The RO can revoke the token to withdraw authorization (consent)

Resource owner

- Authorizes (consents) at run time after authenticating, at the AS
- App gets consent based on the API scopes it requested; it has its own identity distinct from the RO's
- Standard OAuth endpoints for authorization and access token issuance
- Some number of API endpoints that deliver the data or other value-add

Authorization Request

Authorization Grant

Authorization Grant

Authorization Grant

Access Token

Access Token

Access Token

Protected Resource

This can come with a refresh token for renewal without the RO's intervention

Client
OpenID Connect does modern-day federation
It is an OAuth-protected identity API, plus a bit more

- Resource owner = Federation user
- Client = Relying party
- Authorization server = Identity provider ("OpenID provider")
- Resource server

Token endpoint typically delivers an “ID token” similar to a SAML assertion

Standard UserInfo endpoint can be called with an access token to look up identity claims
User-Managed Access is for cross-party sharing

UMA brings next-gen delegation and consent to OAuth

- Ahead of time
- Anytime
- At run time
- After the fact

UX
Share
Monitor
Withdraw
Opt in
Approve

Requesting party

Authorization server

Resource server

Resource server

© 2016 ForgeRock. All rights reserved.
A FinTech use case: Origo solution for the UK Pensions Dashboard project

https://youtu.be/LjWPyy94NgA

oixuk.org/digital-id-for-pensions-dashboards/

WHAT HAVE I GOT AND WHERE IS IT? IDENTITY, ATTRIBUTES AND UMA FOR A PENSIONS DASHBOARD

Kenneth May, Lead Architect, Origo

The Pensions Dashboard project is an important and exciting initiative for the UK consumer with an immense social purpose. It has the potential to significantly improve retirement planning, financial inclusion and consumer engagement with the pensions industry. Origo is working with ForgeRock and the wider industry to bring an enabling infrastructure to market. The solution will securely identify the consumer before orchestrating a search of pensions across the industry. Today we will provide a tour of the project to date. We’ll cover the architecture for identity, attribute exchange and resource sharing; bringing this to life with a demonstration.
First, discovery and aggregation of pension pots
Alice consents to pension pot discovery results with a client she uses
Now she can share with financial advisors
Permissions are granular and revocable
Health use cases: The HEART Working Group at OpenID Foundation

The intent of the Health Relationship Trust (HEART) Working Group is to develop, primarily through profiling, a set of privacy and security specifications that enable an individual to control the authorization of access to RESTful health-related data sharing APIs, and to facilitate the development of interoperable implementations of these specifications by others.

openid.net/wg/heart/
HEART use cases collected

- Multiple portals
- Virtual patient registration
- Post-myocardial infarction implant and rehab
- VA secure RESTful use case
- Patient data for clinical and research purposes
- Primary care physician first appointment
- Alice selectively shares health-related data with physicians and others
Deliverables: All are in Implementer’s Draft status

- HEART Profile for UMA*
- HEART Profile for OAuth 2.0
- HEART Profile for OpenID Connect
- HEART Profile for OAuth 2.0 and FHIR
- HEART Profile for UMA* and FHIR
- HEART Profile for OAUTH-RELATED
- SECURITY PROFILES
- SEMANTIC PROFILES
Demonstration

Consumer/clinical health IoT scenario making use of device attestation and Identity Relationship Management
From the UMA2 Recommendations…

The UMA extension grant spec enhances OAuth in the following ways:

- The resource owner authorizes protected resource access to clients used by entities that are in a requesting party role. This enables party-to-party authorization, rather than authorization of application access alone.

- The authorization server and resource server interact with the client and requesting party in a way that is asynchronous with respect to resource owner interactions. This lets a resource owner configure an authorization server with authorization grant rules (policy conditions) at will, rather than authorizing access token issuance synchronously just after authenticating.

The (optional) federated authorization spec enhances the UMA grant as follows:

- This specification extends and complements [UMAGrant] to loosely couple, or federate, its authorization process.

- This enables multiple resource servers operating in different domains to communicate with a single authorization server operating in yet another domain that acts on behalf of a resource owner.

- A service ecosystem can thus automate resource protection, and the resource owner can monitor and control authorization grant rules through the authorization server over time. Further, authorization grants can increase and decrease at the level of individual resources and scopes.
Sample UMA Grant flow as it appears in the spec

See also:

tinyurl.com/uma2grantwsd
tinyurl.com/uma2fawsd
Key benefits of UMA to consumers

Constrained party-to-party delegation

Granting consent without external influence

Centralized monitoring and management

Control of consents at a fine grain
Key benefits of UMA to service providers

- True security of delegated access
- Scalability of resource permissioning
- API-first protection strategy
- Fosters control for compliance and trust
The “BLT sandwich”

Putting together a business, legal, and technical framework for UMA usage and deployment
The UMA Legal effort
tinyurl.com/umalegal

• A subgroup of the UMA Work Group, with expert legal help from Tim Reiniger, has produced a draft framework defining:

  How the UMA protocol enables a license-based model for controlling access rights to personal digital assets

• The business tools we (and others) can build from this framework can be single- and cross-jurisdictional, and single- and cross-sector

• Our contention:

  UMA can provide the autonomy, reciprocity, and objectivity to grow market trust in widely sharing access to personal digital assets with devices, apps, and Internet databases.
Legal roles and artifact interactions
Delegation and licensing: RO-centered

- Delegates authorization for granting access permissions
  - Data Subject
  - Resource Owner
  - Authorization Server Operator
  - UMA artifacts: none

- Delegates management of resources
  - Data Subject
  - Resource Owner
  - Resource Server Operator
  - UMA artifacts: none

- Licenses granting access permissions on Resource Owner’s behalf
  - Authorization Server Operator
  - Resource Server Operator
  - Limited; Resource Server can only restrict access further, not give more access
  - UMA artifacts: Resource Server’s OAuth client credentials, PAT (with Resource Owner context), all request/response messages

Key:
- Legal devices only
- Bound to UMA artifacts
Delegation and licensing: receiving permissions

Example message set:
Client can revoke RPT to withdraw granted access permissions on Requesting Party’s behalf

UMA artifacts: Client’s OAuth client credentials, RPT (with permissions), claim token, all request/response messages

UMA artifacts: RPT (with permissions), claim token, all request/response messages
Delegation and licensing: RqP-centered

Delegates access seeking

Requesting Party

In a Limited Agent role

Client Operator

UMA artifacts: claim token, PCT, all request/response messages

Delegates permission to know/persist

Requesting Party

Authorization Server Operator

UMA artifacts: PCT, all request/response messages

Key:

Partially bound to UMA artifacts
To which GDPR articles is UMA most relevant?

- **Article 5**: Principles relating to personal data processing
- **Article 7**: Conditions for consent
- **Article 8**: Conditions applicable to child’s consent in relation to information society services
- **Articles 12-23**: Rights of the data subject
- **Article 25**: Data protection by design and by default
  - Right of access, to restrict processing, to withdraw consent at any time
  - User-controlled delegation of access
  - Centralizable management of consents
  - Transparency of and control over consented access
  - Consented sharing directly from sources for greater data accuracy
  - Parent/guardian-to-child constrained delegation
What changed from UMA1 to UMA2 and what’s coming next
Key delta: The AAT is gone

- It got in the way of just-in-time wide-ecosystem trust elevation of requesting parties the AS had never met before
  - Though we don’t say “trust elevation” anymore…

- We removed the requirement to use OAuth for the AAT's previous job, and now offer an optional PCT (persisted claims token) to capture results of interactive claims gathering, noting that the AS could give the RqP a chance to authorize its issuance
Key delta: The AAT is gone

- Now, if you’re using a federated login flow for the RqP, the first time they have to do anything, it’s squarely in the realm of interactive claims gathering.
GitHub issues with extension ideas
Thank you!
Questions?
Join us!

Eve Maler
VP Innovation & Emerging Technology, ForgeRock
@xmlgrrl | eve.maler@forgerock.com
Chair and founder, Kantara UMA Work Group
@UMAWG | tinyurl.com/umawg
24 Jan 2018