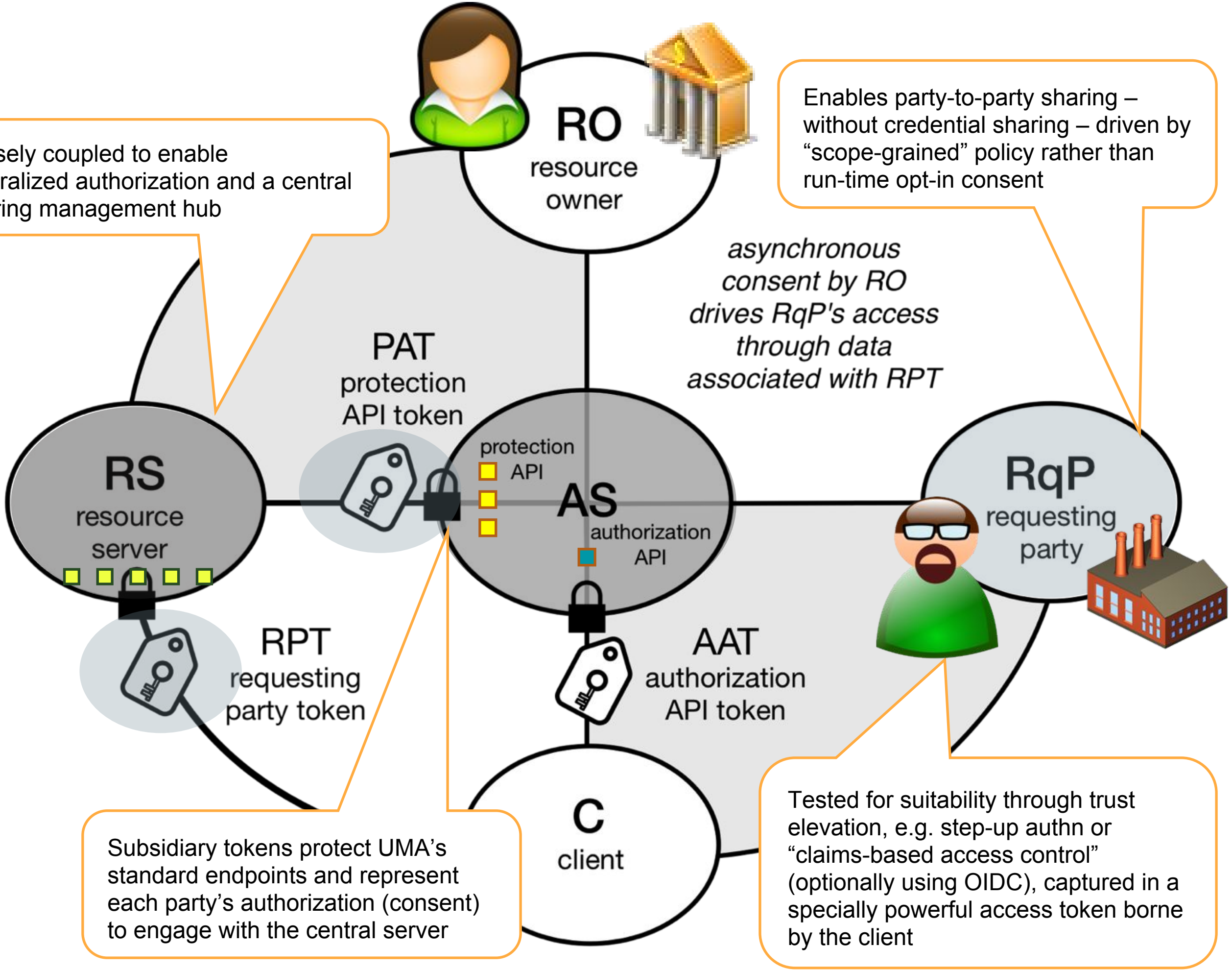
Status Report:
UMA WG (including Legal)
Blockchain and Smart Contracts DG

Eve Maler | @xmlgrrl
9 May 2017

UMA WG

UMA1 added party-to-party, asynchronous, scope-grained delegation and control to OAuth

tinyurl.com/umawg



Loosely coupled to enable centralized authorization and a central sharing management hub

Enables party-to-party sharing – without credential sharing – driven by "scope-grained" policy rather than run-time opt-in consent

Subsidiary tokens protect UMA's standard endpoints and represent each party's authorization (consent) to engage with the central server

Tested for suitability through trust elevation, e.g. step-up authn or "claims-based access control" (optionally using OIDC), captured in a specially powerful access token borne by the client

UMA 2.0 themes

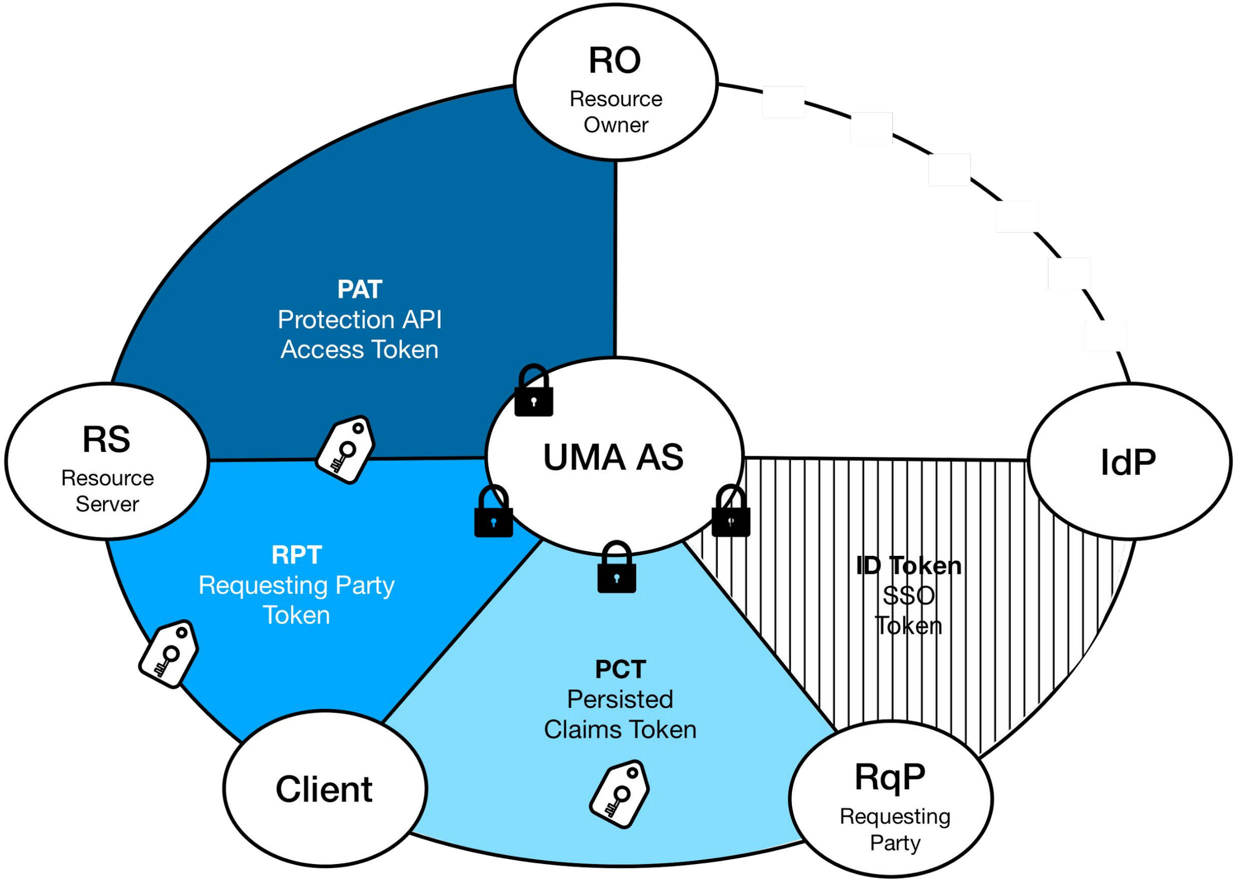
- Align more closely to OAuth
  - To accelerate adoption and interop
- Improve suitability for IoT scenarios
- Improve suitability for “wider” ecosystems
  - Requesting parties less known to/controlled by the authorization server

- The solutions to these challenges turned out to be intertwined...and we believe UMA2 has feature parity

UMA 2.0 timeline

- Dec 2015: UMA V1.0.1 Recommendations published
- Q1 2016: 2.0 roadmap themes discussed and decided
- Early Q2 2016: Major decision-making begun
- May 2016: Spec editing begun
- Jan 2017: Completed editing of key “jricher” design issues
- Mar 2017: Completed editing of follow-on issues
- Apr 2017: Completed spec refactoring
- 12 May 2017 (**this Friday**): Planned WG vote on closing remaining follow-on issues and starting Public Comment and IPR Review period

UMA2 achieves UMA1’s aims and more, through an **extension OAuth grant** and **optional federated authorization**



Specs and benefits

UMA Grant

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.

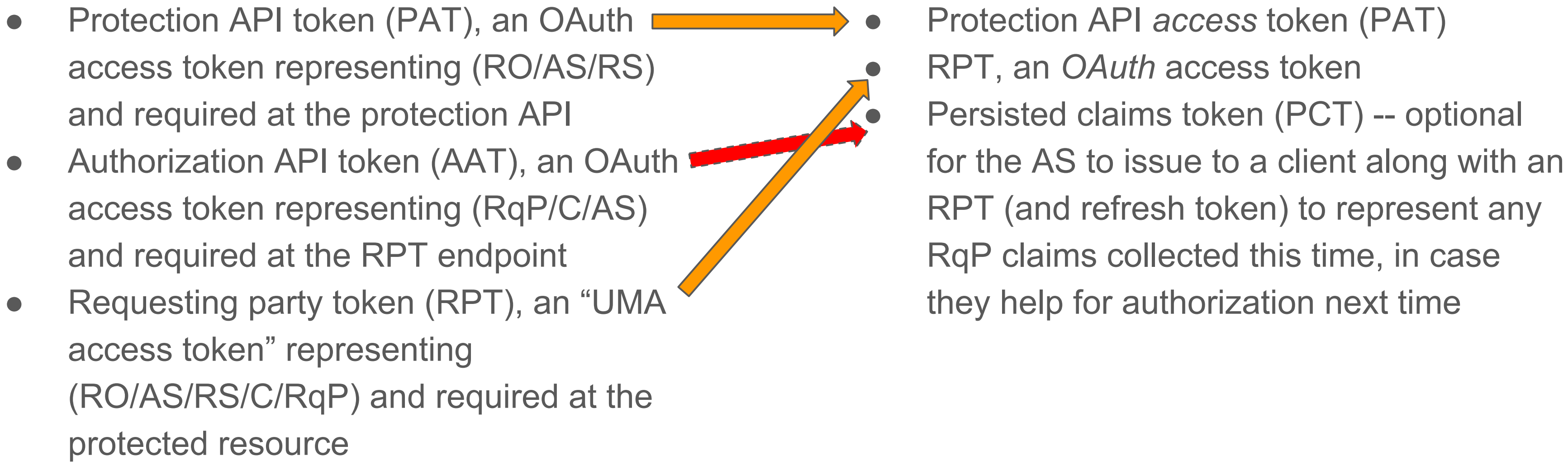
Federated Authorization for UMA

...Loosely couple[s], or federate[s], the 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** at a central service location over time. Further, with the use of token introspection, authorization grants can **increase and decrease at the level of individual resources and scopes**.

Language clarification and alignment

UMA1	UMA2
resource set registration, resource set	resource registration, resource (protected while registered)
authorization API	UMA grant *(an extension OAuth grant)*
register a permission (for permission ticket)	request *(one or more)* permission(s) (on behalf of a client)
“policies” *(colloquial)*	access grants, access grant rules, policy conditions
trust elevation	authorization process *and* authorization assessment
claims pushing + claims gathering = *(n/a)*	claims pushing + claims gathering = claims collection
step-up authentication	*(n/a)*; *just* authorization process
authorization API token (AAT)	*goes away; a new related token is* persisted claims token (PCT)
RPT *as an UMA access token*	RPT *as an OAuth access token*
protection API token (PAT)	protection API access token (PAT)

Tokens



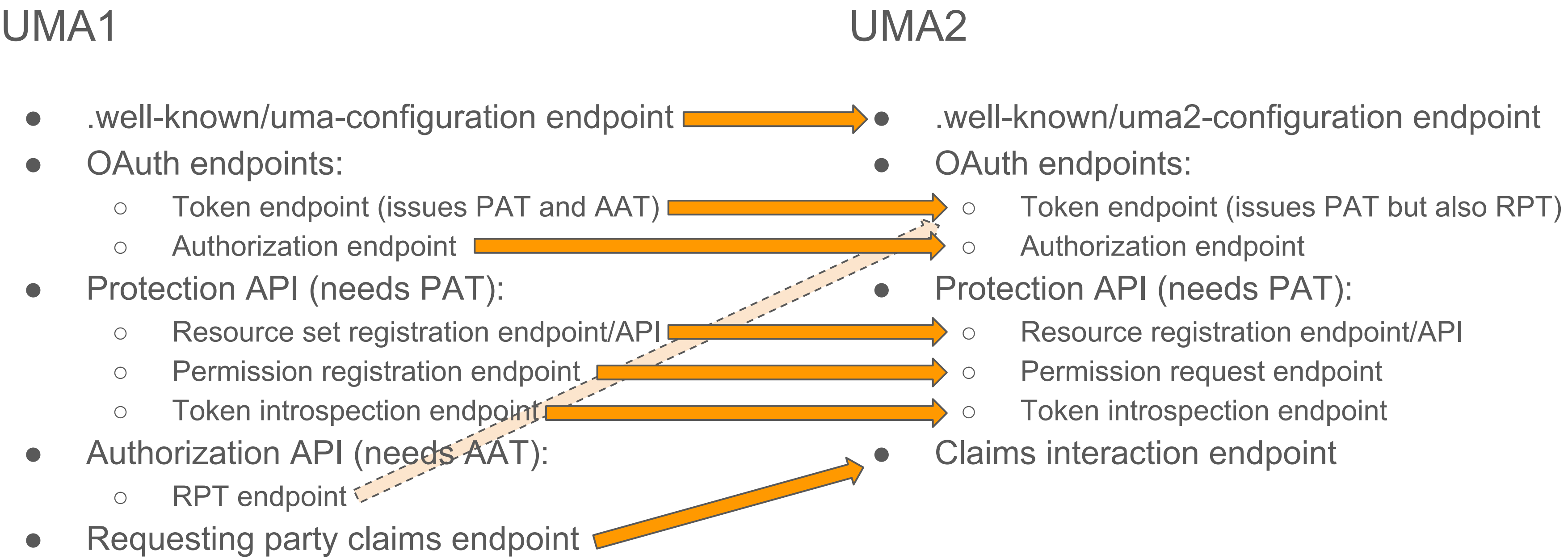
UMA1

- Protection API token (PAT), an OAuth access token representing (RO/AS/RS) and required at the protection API
- Authorization API token (AAT), an OAuth access token representing (RqP/C/AS) and required at the RPT endpoint
- Requesting party token (RPT), an “UMA access token” representing (RO/AS/RS/C/RqP) and required at the protected resource

UMA2

- Protection API *access* token (PAT)
- RPT, an *OAuth* access token
- Persisted claims token (PCT) -- optional for the AS to issue to a client along with an RPT (and refresh token) to represent any RqP claims collected this time, in case they help for authorization next time

APIs and endpoints



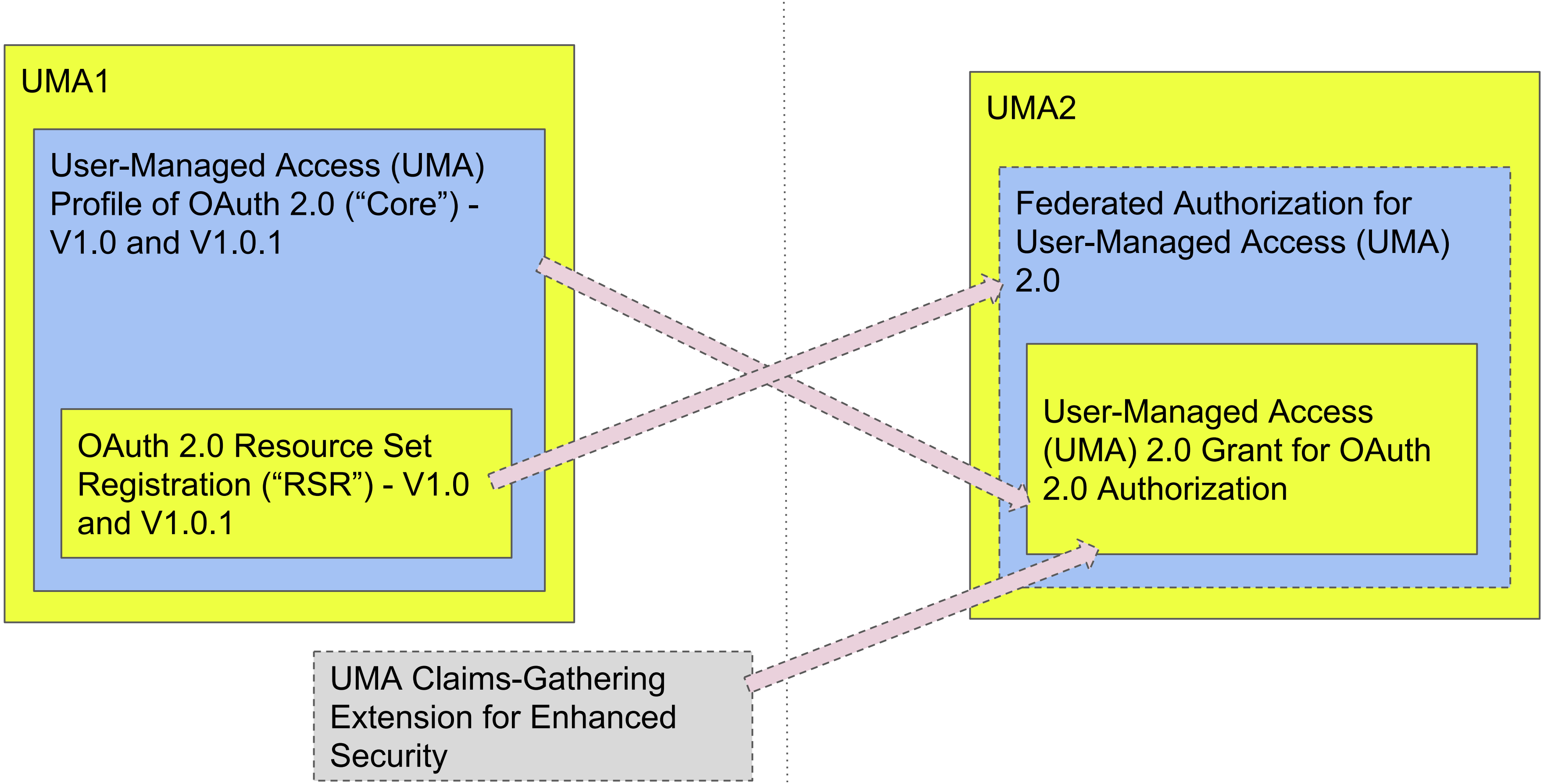
UMA1

- .well-known/uma-configuration endpoint
- OAuth endpoints:
  - Token endpoint (issues PAT and AAT)
  - Authorization endpoint
- Protection API (needs PAT):
  - Resource set registration endpoint/API
  - Permission registration endpoint
  - Token introspection endpoint
- Authorization API (needs AAT):
  - RPT endpoint
- Requesting party claims endpoint

UMA2

- .well-known/uma2-configuration endpoint
- OAuth endpoints:
  - Token endpoint (issues PAT but also RPT)
  - Authorization endpoint
- Protection API (needs PAT):
  - Resource registration endpoint/API
  - Permission request endpoint
  - Token introspection endpoint
- Claims interaction endpoint

Spec refactoring



UMA1

User-Managed Access (UMA) Profile of OAuth 2.0 (“Core”) - V1.0 and V1.0.1

OAuth 2.0 Resource Set Registration (“RSR”) - V1.0 and V1.0.1

UMA2

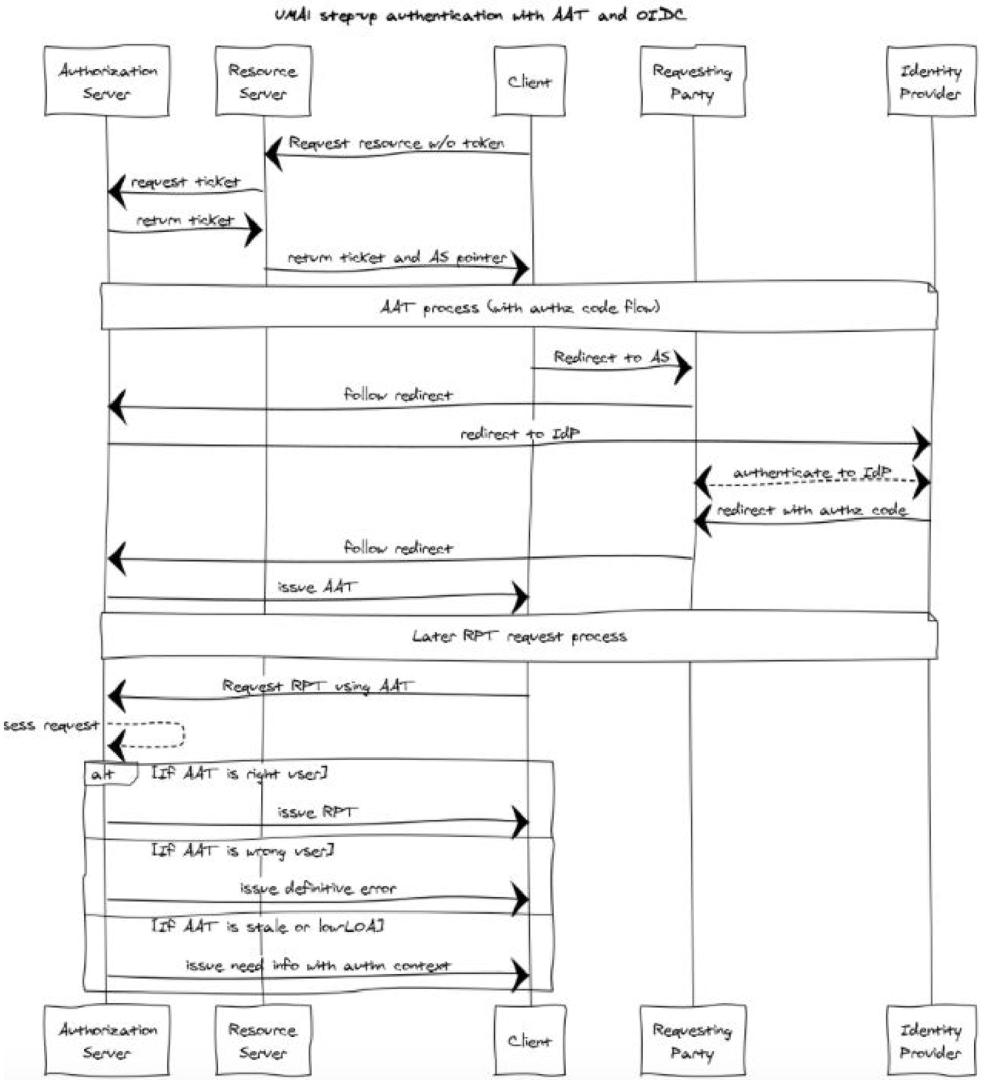
Federated Authorization for User-Managed Access (UMA) 2.0

User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization

UMA Claims-Gathering Extension for Enhanced Security

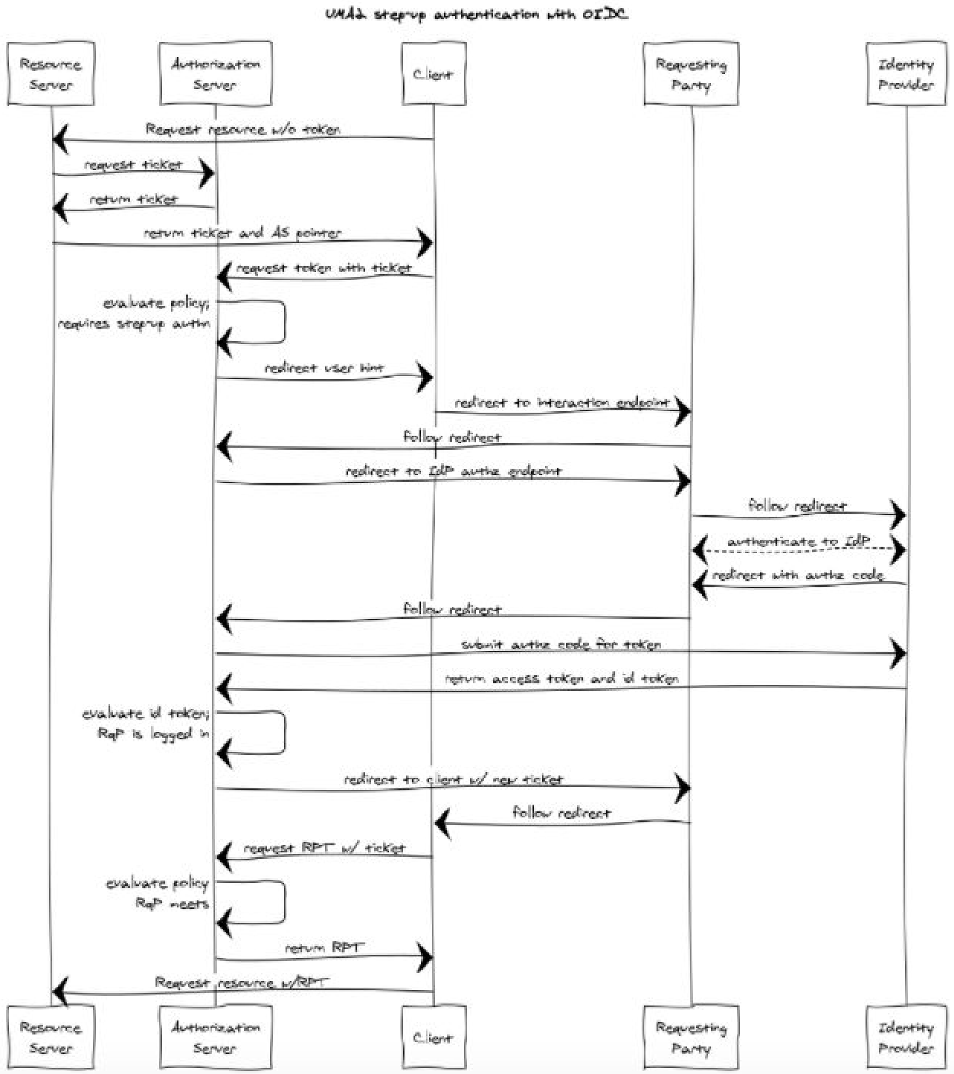
UMA1 AAT flow using OIDC login

- Requesting party *must* go through an in-band OAuth(-based) flow, “eagerly”
- This *requires* an identity-specific interaction
- What if the resource owner’s policy said only “Share with anyone over 18?”
- See the original swimlane here



UMA2 federated login as a claims-gathering flow

- Requesting party is just redirected to the authorization server when seeking access
- Authorization server is an RP/claims client
- See the original swimlane here


UMA2 stepup authentication with OIDC
Resource Server
Authorization Server
Client
Requesting Party
Identity Provider
Request resource w/o token
request ticket
return ticket
return ticket and AS pointer
request token with ticket
evaluate policy; requires stepup authn
redirect user hint
redirect to interaction endpoint
follow redirect
redirect to IdP authz endpoint
follow redirect
authenticate to IdP
redirect with authz code
follow redirect
submit authz code for token
return access token and id token
evaluate id token; RqP is logged in
redirect to client w/ new ticket
follow redirect
request RPT w/ ticket
evaluate policy RqP meets
return RPT
Request resource w/RPT
Resource Server
Authorization Server
Client
Requesting Party
Identity Provider

UMA Legal Subgroup

Mission

- Produce toolkits and educational materials by end of 2017 to accelerate adoption, deployment, and use of UMA-enabled services in a manner consistent with protecting privacy rights
    - A toolkit could be an SDK, a checklist, consent receipt templates or profiles, or CommonAccord text, and could be related to the GDPR itself, the EU-U.S. Privacy Shield, BCRs, and so on
    - Focus on GDPR-related toolkits first and foremost
- To inform this work, develop a legal framework through use cases and analysis, leveraging specialist legal expertise
    - E.g., bridge UMA concepts and regulatory concepts such as “data subject”, “data processor”, “data controller”
- Framework is being developed and delivered by Tim Reiniger in three installments

Deliverable #1 was completed 28 Feb ‘17

- First part: *Lex Informatica* considerations
  - UMA value proposition, UMA consent advantages, UMA legal advantages, UMA legal challenges
- Second part: Salient factors for use cases
  - Networked-access environments, Resource Subject/Owner variations, Resource Server variations, Requesting Party variations, authorization permissions/purposes
- Third part: UMA use cases by networked-access environment
- Fourth part: Implications for creating an UMA legal framework

Deliverable #2 completion is anticipated by May 12

- It includes cross-products of functional, liability, and legal implications for UMA access-granting relationships from both the Resource Owner perspective and the Requesting Party perspective
- We are proceeding on the assumption that the correct legal model is *licensing*

Deliverable #3, the legal framework, will follow

Blockchain and Smart Contracts Discussion Group

Co-chairs: Eve Maler and Thomas Hardjono

BSC DG timeline

- 5 Jul 2016: DG launched with a six-month completion timeframe
- Jul 2016: DG quickly crisped up its area of inquiry: *analyzing novel attempts to use blockchain and distributed ledger technologies to achieve an equitable distribution of accountability and risk: what could be described as "personal data and transaction ecosystems in which individuals and organizations can interact more equitably and efficiently"*
- 5 Jan 2016: DG agreed to keep working on its draft Report
- 5 May 2016 (*"January 125th"*): DG achieved consensus to wrap up its Report (except for minor copy-editing) and deliver it to Kantara Initiative

Technologies and techniques included in the report

- Blockchains and Distributed Ledger Technologies (DLTs)
- Legal Contracts and Smart Contracts
- InterPlanetary File System (IPFS) & Content Based Networks
- Certificate Transparency
- Verifiable Claims
- OPAL/Enigma
- Protocol-Specific Contract Provisions
- CommonAccord
- User-Managed Access (UMA)
- Consent Receipts
- User Submitted Terms
- Identity and Access Management

Use cases included in the report

- Personal Health Information for Research Purposes
- Sovrin-Based Self-Sovreign Identity
- Alice Participates in Bob’s Research Study
- Research Evidence Notebook
- Smart Medical Telematics
- Prescription Writing Into a Patient’s Health Record

Recommendations provided in the report

- Launch a Blockchain and Smart Contracts WG
- Consider a Kantara-Wide Legal WG
- Research Inside and Outside Kantara

Many thanks to the tireless DG participants, and special thanks to Thorsten Niebuhr and his contributors from the IRM and IDPro groups who helped us with content!