

The State of Kantara User-Managed Access (UMA) Version 1.0


Eve Maler, chair (@xmlgrrl)

Maciej Machulak, vice-chair (@mmachulak)

@UMAWG | tinyurl.com/umawg

16 May 2015





UMA v1.0 Calling for Implementations

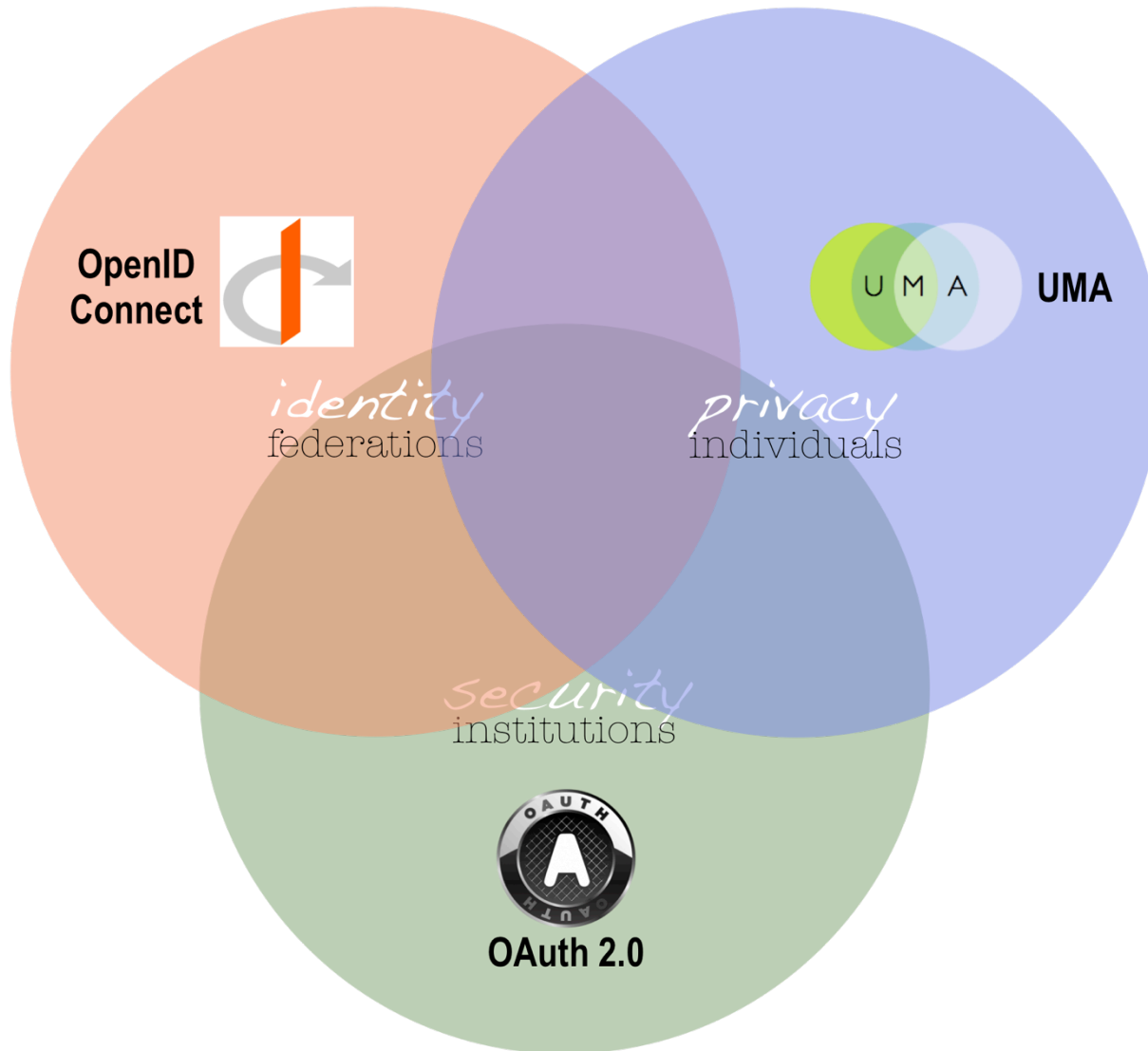
Kantara UMA Standard Achieves V1.0 Status, Signifying A Major Milestone for Privacy and Access Control. Kantara Initiative is calling on organizations to implement User-Managed Access in applications and IoT systems.

[Get Involved](#)

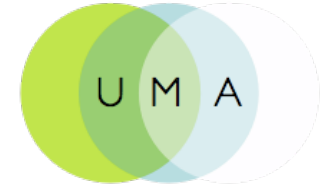
The big picture



The new Venn of access control and consent



UMA in a nutshell

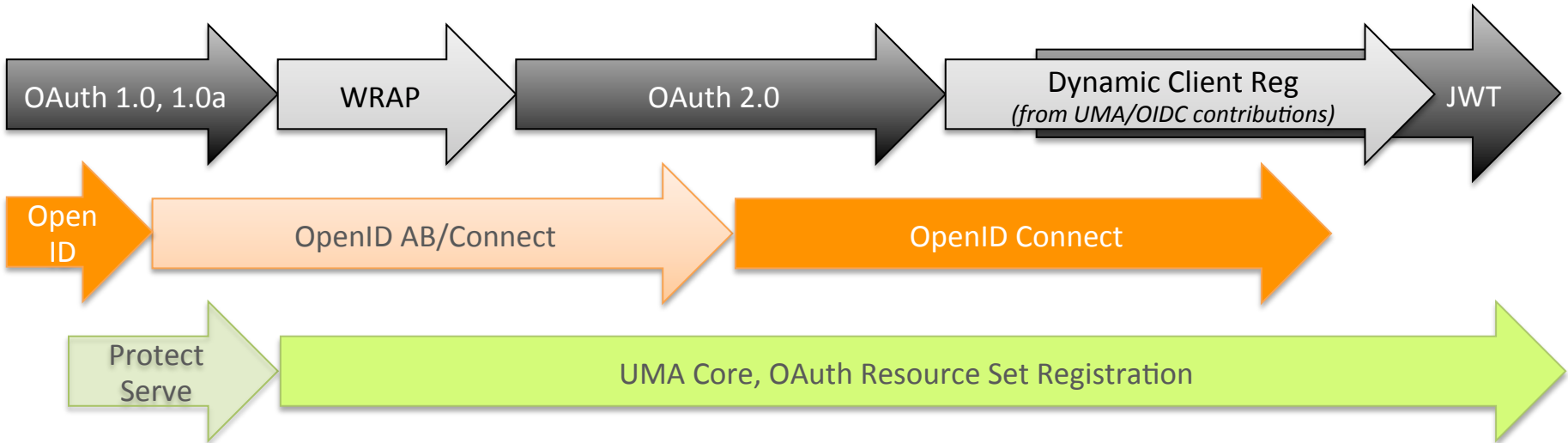


- It's a protocol for lightweight access control
- It's a profile and application of OAuth2
- It's a set of authorization, privacy, and consent APIs
- It's a Kantara Initiative Work Group
- It's made up of two V1.0 Recommendations (standards)

Standardization progress in context



08 09 10 11 12 13 14 15

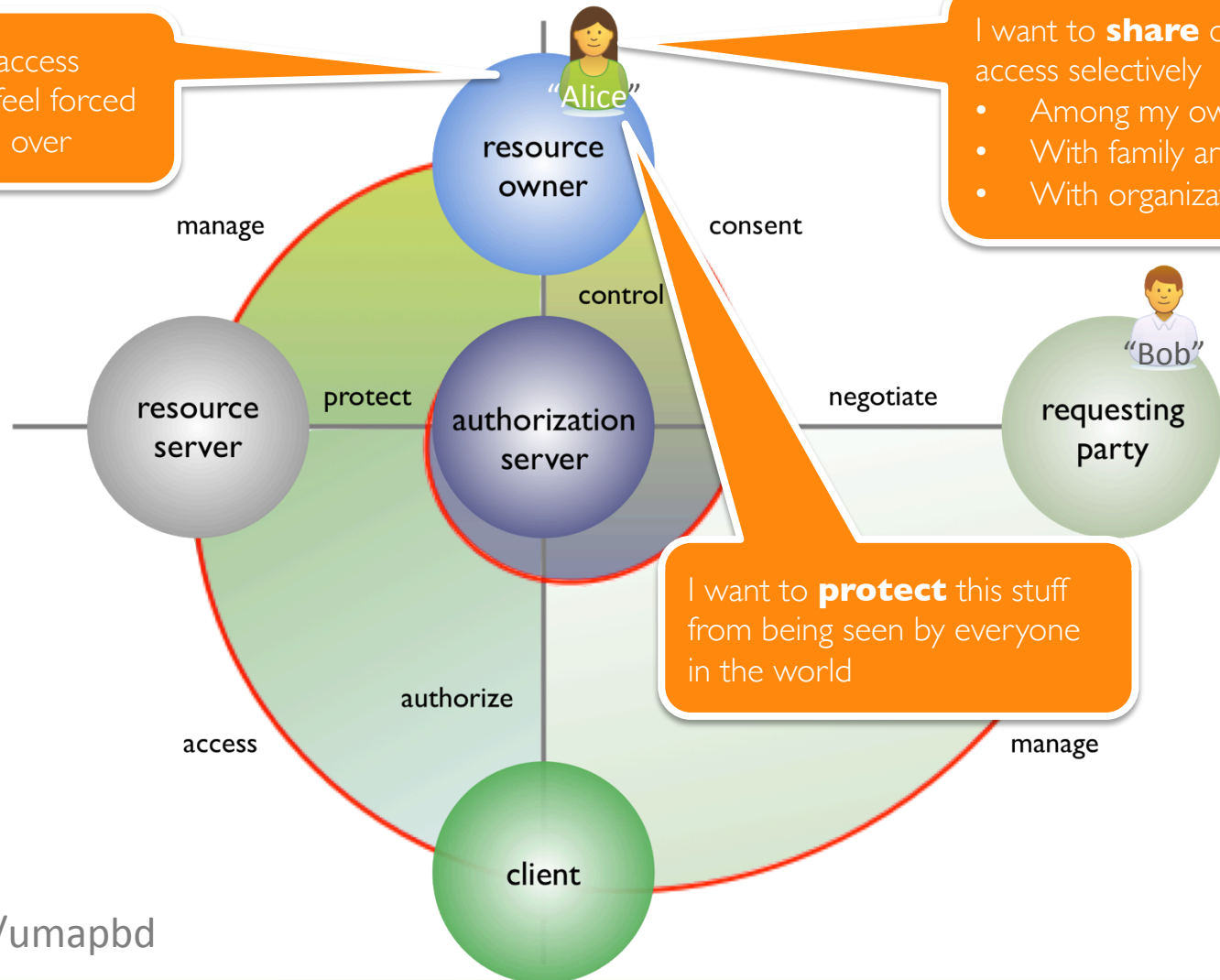


UMA brings to Alice's interactions with services and net-connected things

I want to **control** access proactively, not just feel forced to consent over and over

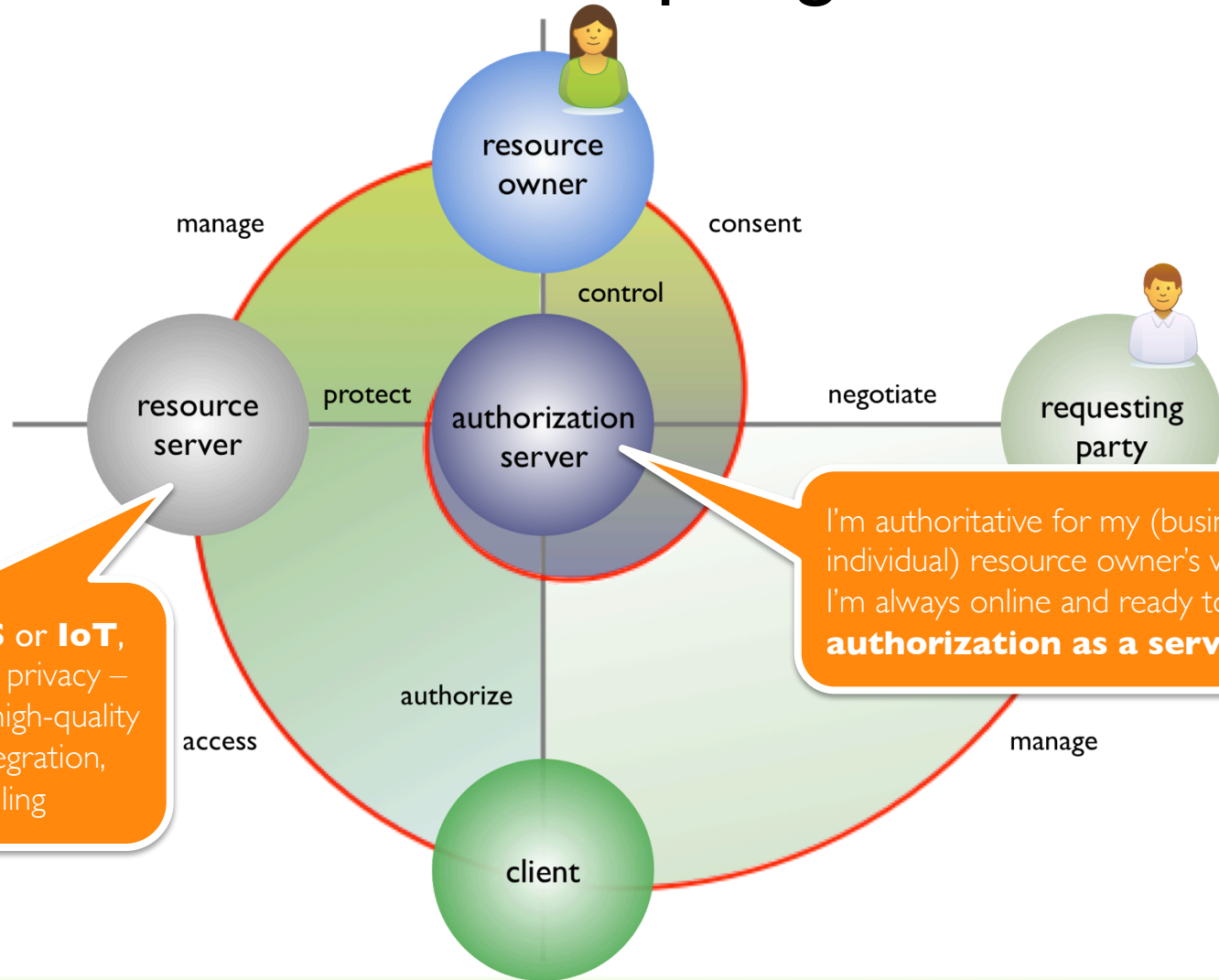
I want to **share** data and access selectively

- Among my own apps
- With family and friends
- With organizations



I want to **protect** this stuff from being seen by everyone in the world

UMA lets apps and services gain high-quality authorization through loose coupling



Use-case domains

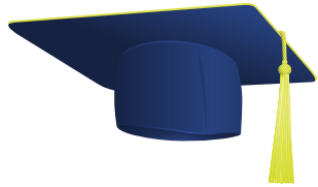


Health

Financial



Education



Personal



Government



Media



Enterprise



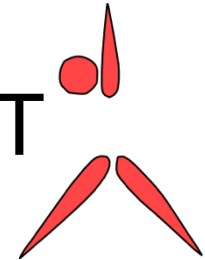
Web

Mobile

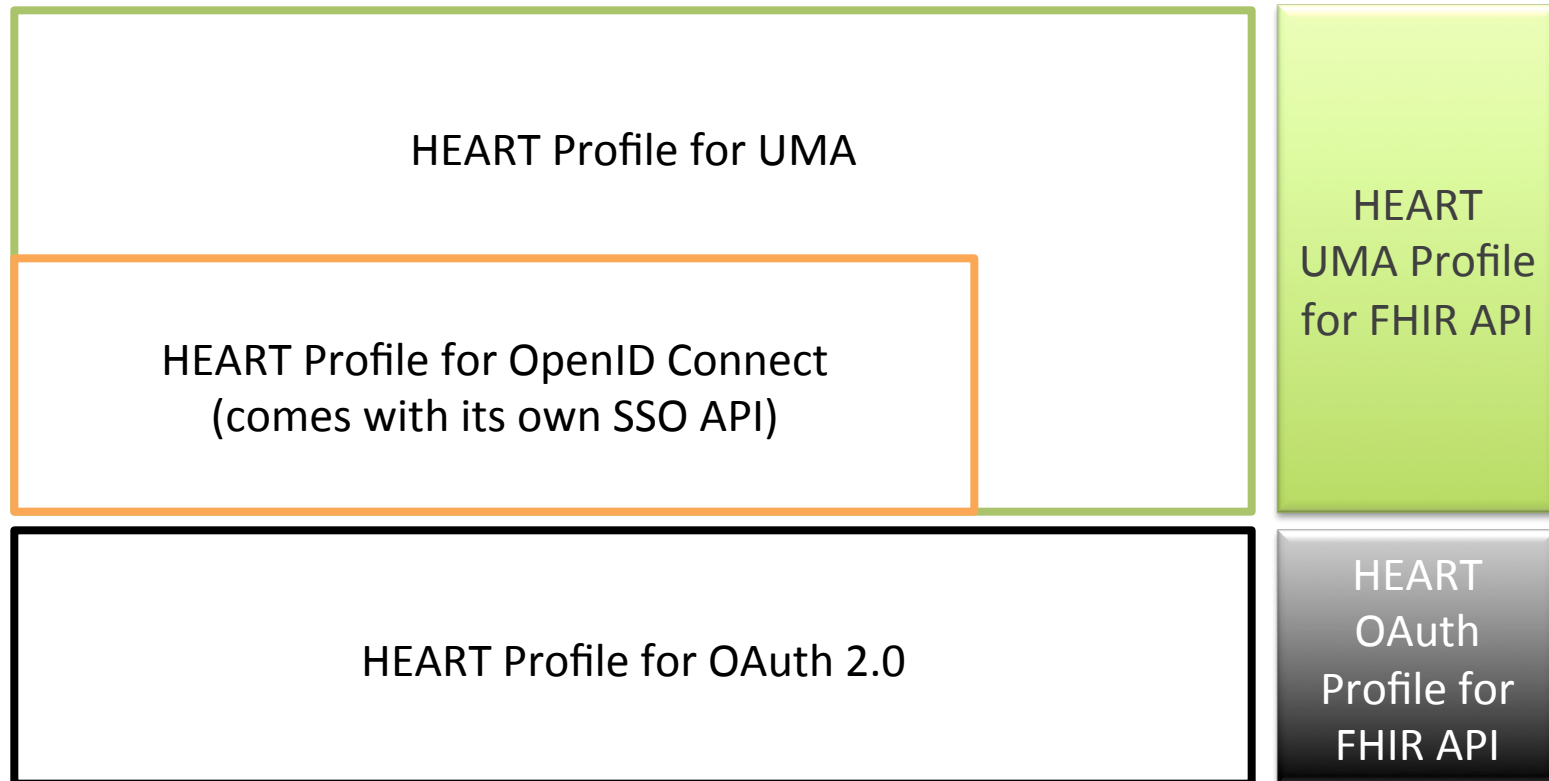


API

IoT



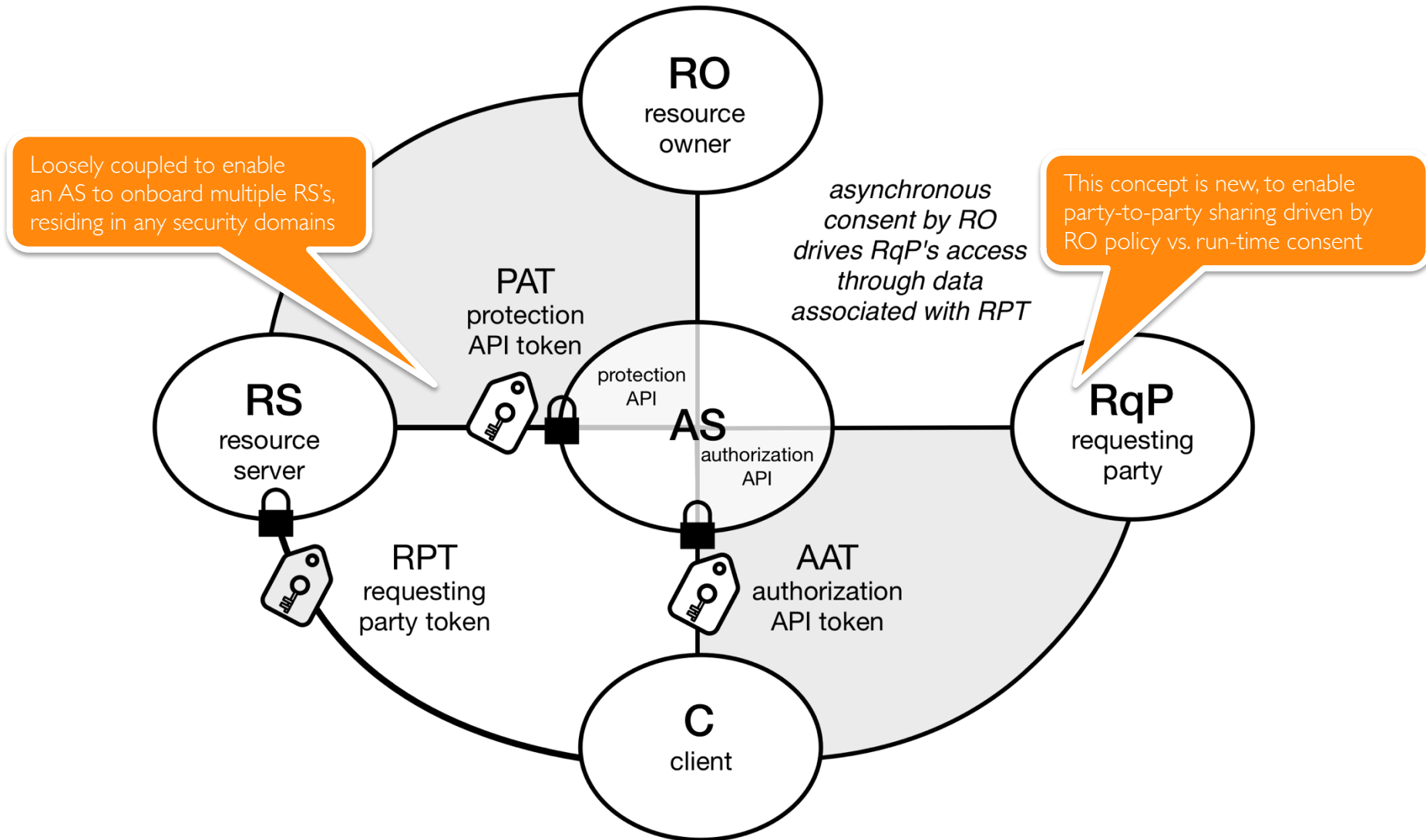
Health Relationship Trust (HEART) in OIDF has an UMA connection



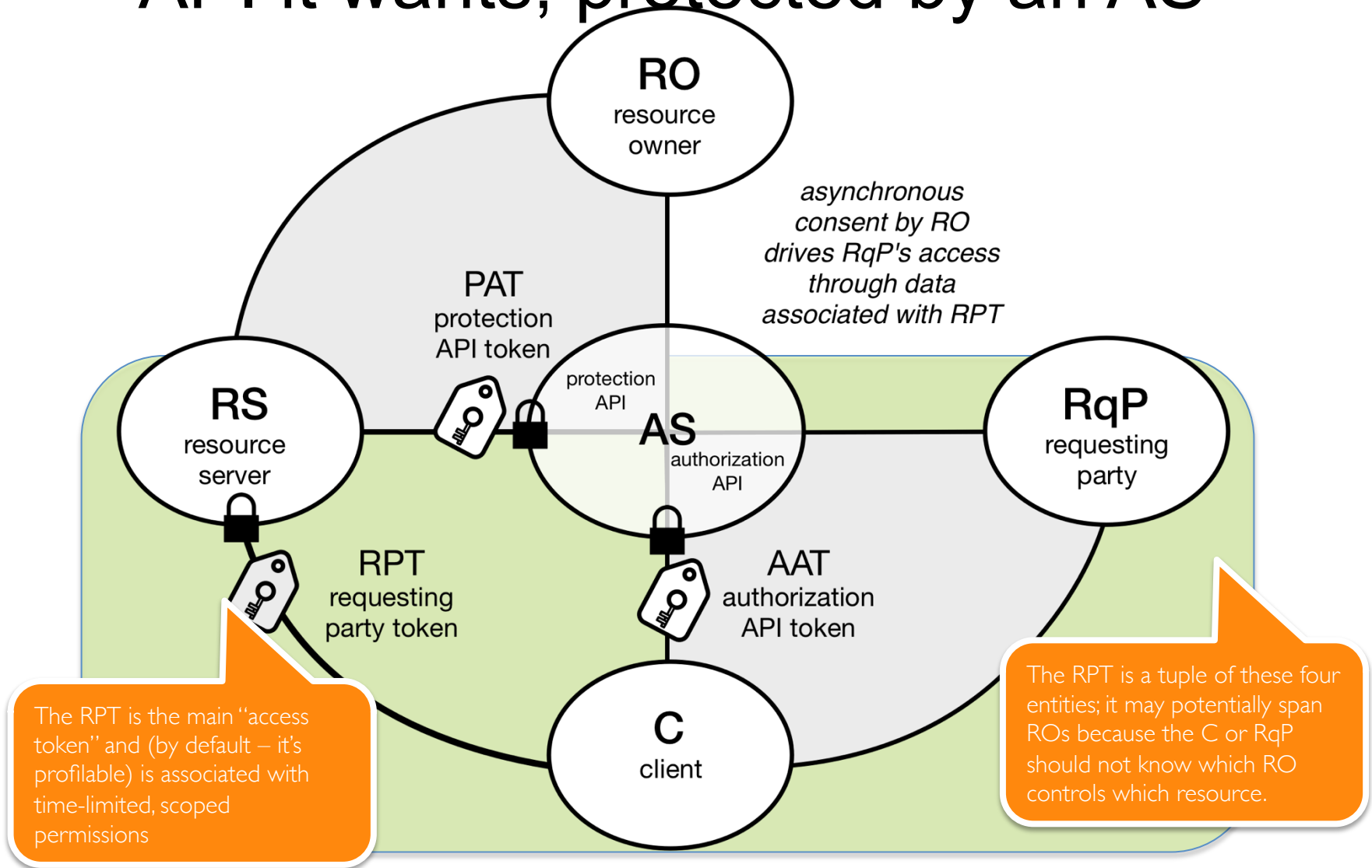
UMA particulars



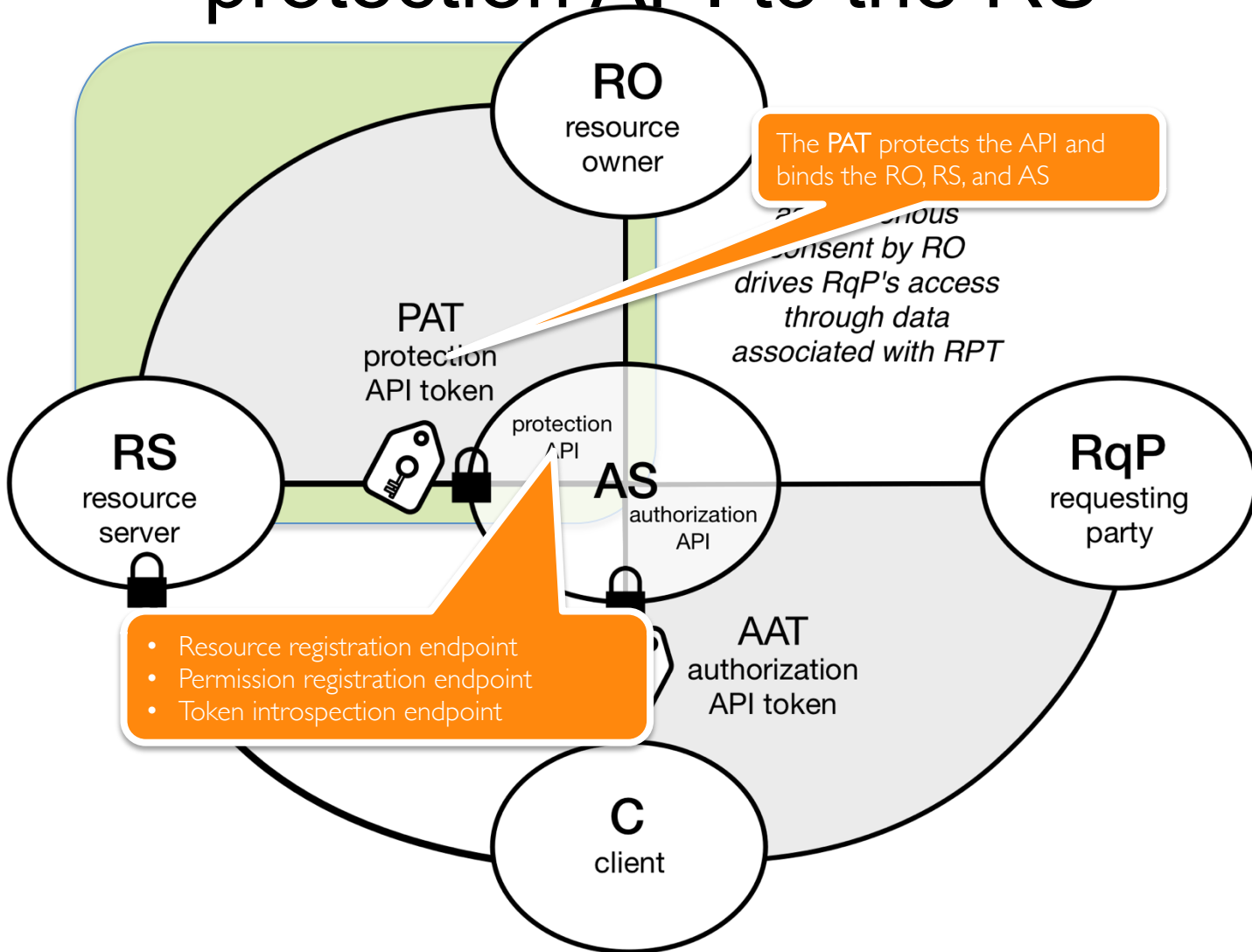
Under the hood, it's "OAuth++"



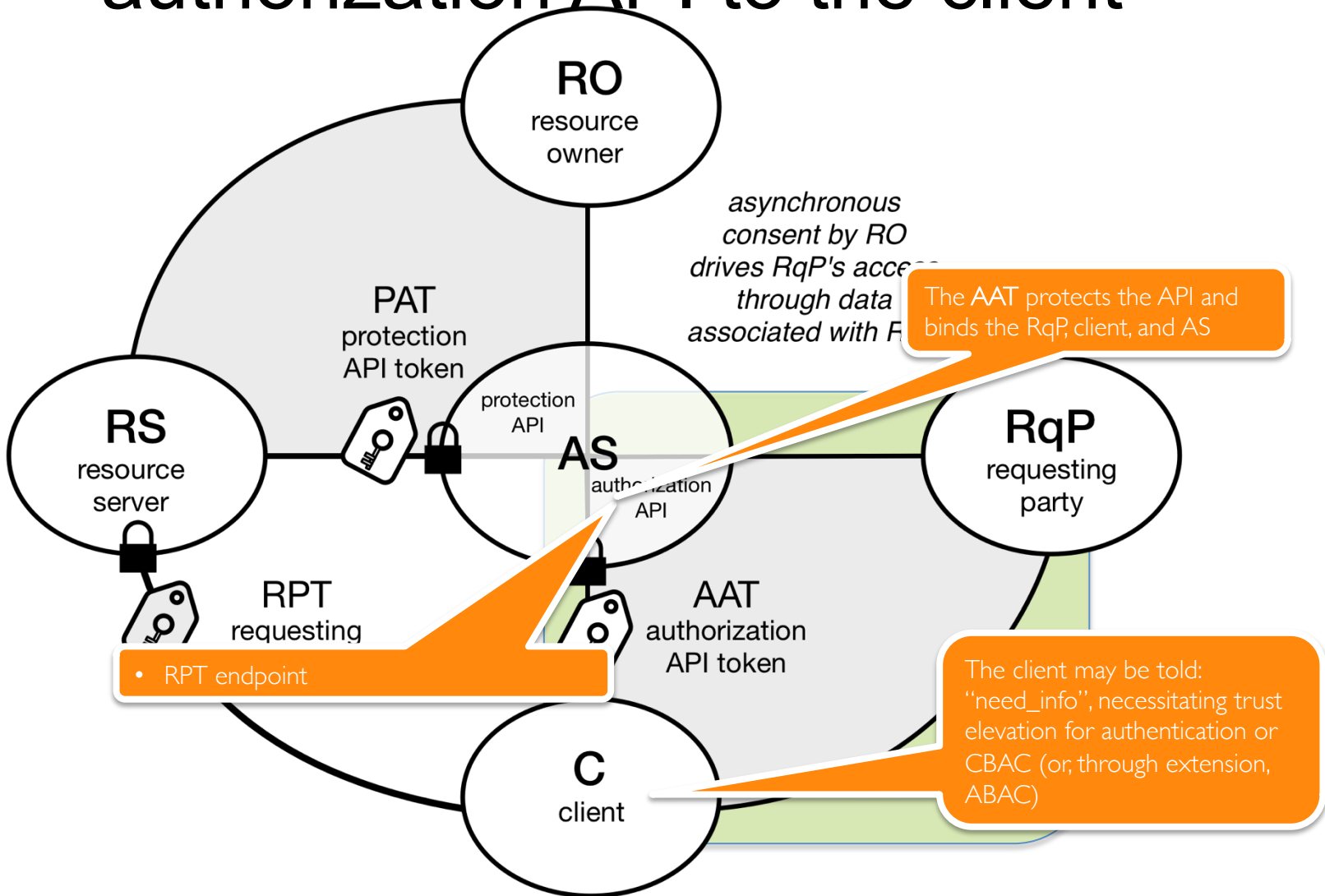
The RS exposes whatever value-add API it wants, protected by an AS



The AS exposes an UMA-standardized protection API to the RS



The AS exposes an UMA-standardized authorization API to the client



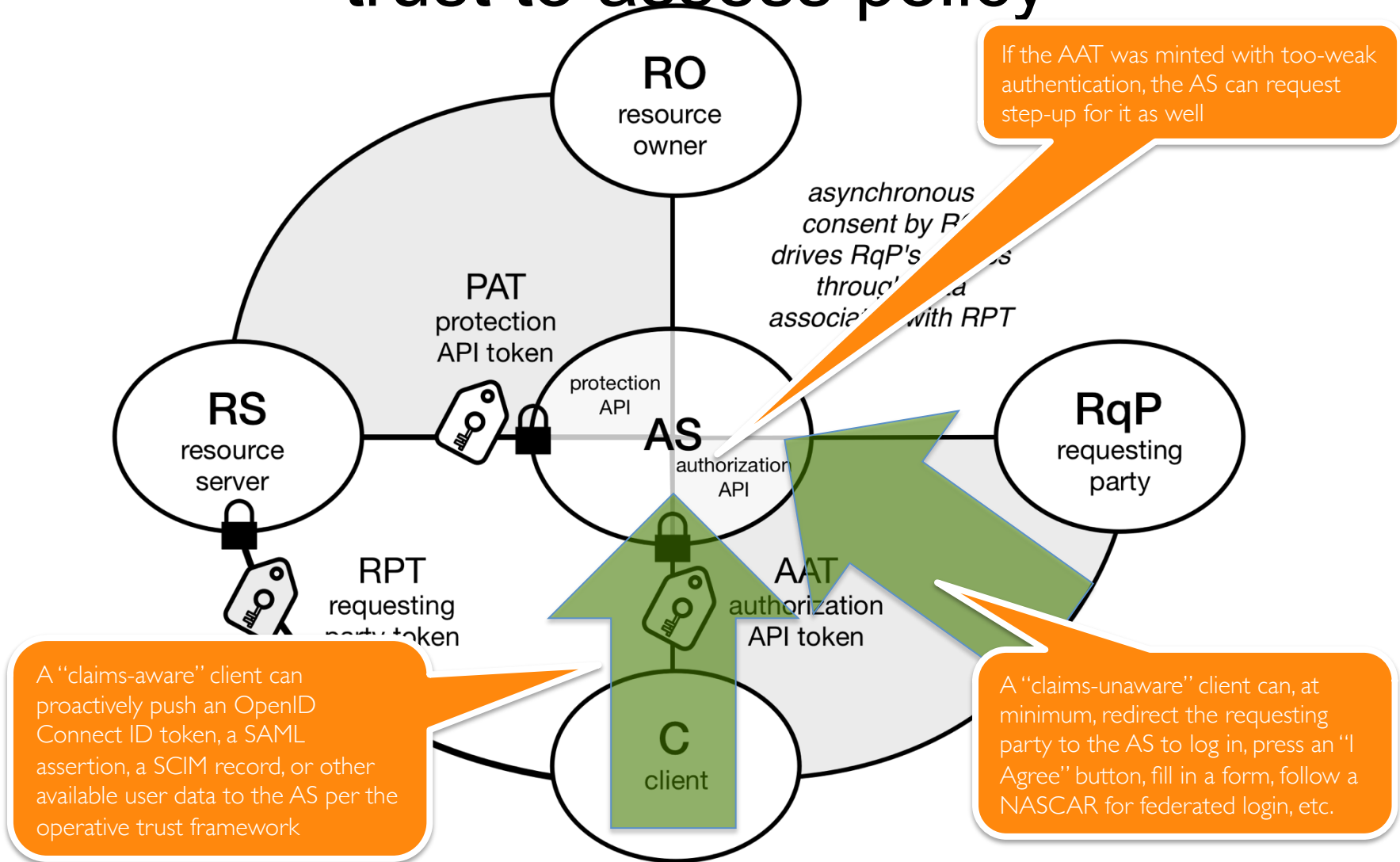
These are *embedded* OAuth flows to protect UMA-standard security APIs

- The “PAT” and “AAT” are our names for plain old OAuth tokens – representing important UMA concepts!
 - Alice’s consent to federate authorization
 - Bob’s consent to share claims to win access
- Many “binding obligations” will hinge on their issuance

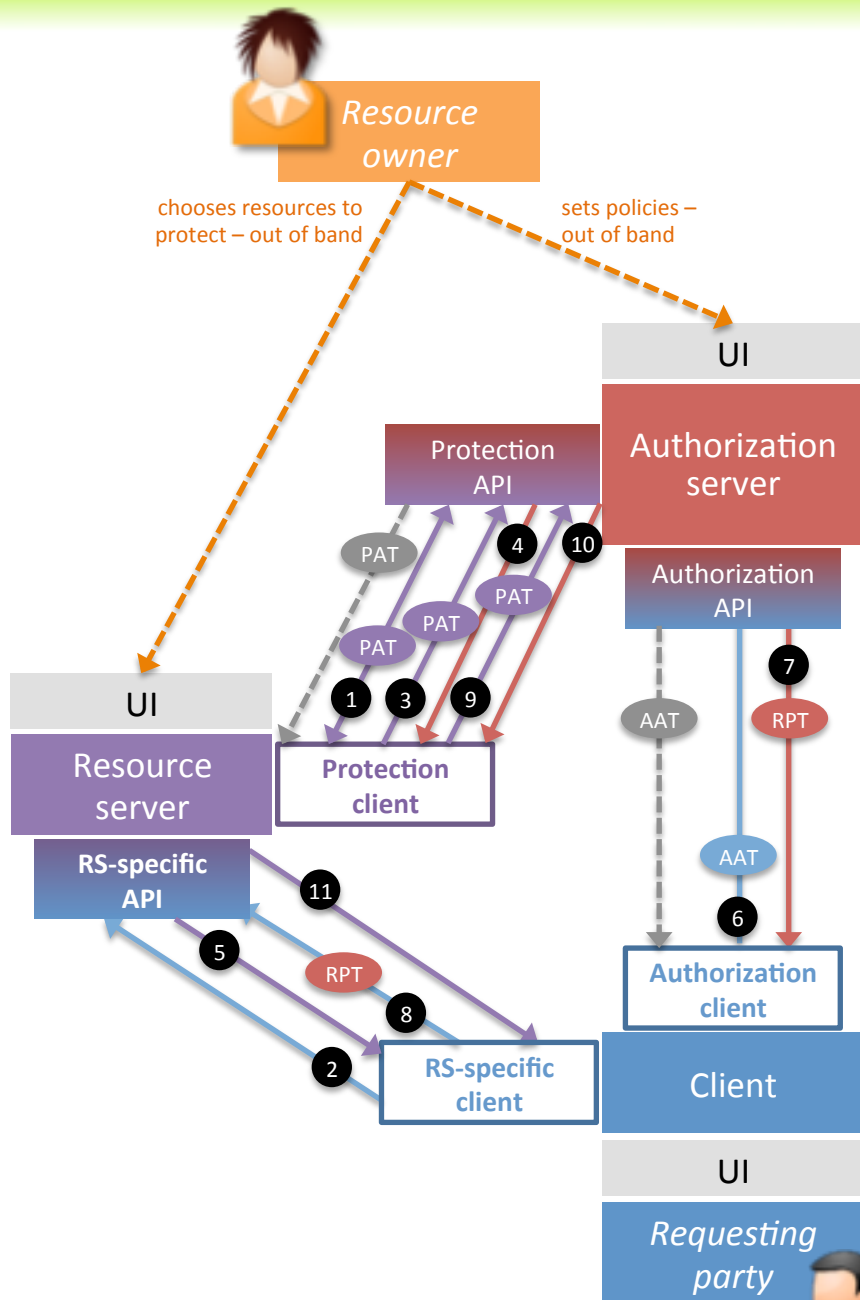
The significance of resource set registration

- The AS is authoritative for Alice's policy
- But the RS is authoritative for what its API *can do* – its “verbs” and “objects”, and what Alice has created there
- Resource set registration allows the RS to remain authoritative in this fashion, and allows RS:AS to be an *n:m* relationship

The AS can elevate requesting party trust to assess policy



High-level UMA flow



RS needs OAuth client credentials at AS to get PAT
 C needs OAuth client credentials at AS to get AAT
 All protection API calls must carry PAT
 All authorization API calls must carry AAT

1. RS registers resource sets and scopes (ongoing – CRUD API calls)
2. C requests resource (provisioned out of band; must be unique to RO)
3. RS registers permission (resource set and scope) for attempted access
4. AS returns permission ticket
5. RS returns error 403 with as_uri and permission ticket
6. C requests authz data, providing permission ticket
7. (After claims-gathering flows not shown) AS gives RPT and authz data
8. C requests resource with RPT
9. RS introspects RPT at AS (if using default “bearer” RPT profile)
10. AS returns token status
11. RS returns 20x



What notably changed from earlier drafts to V1.0?

- Resource set registration:
 - Scopes can now be plain strings instead of strictly URIs that resolve to JSON descriptions
 - Create went from PUT with RS-assigned ID to POST with AS-assigned ID
 - Can register a “uri” resource set location for usage in discovery
- Core:
 - Simplified the RPT issuance flow and removed the dedicated RPT issuance endpoint; permissions are now also registered eagerly
 - Massively upgraded the trust elevation capability (now called “need_info”) to handle both claims-gathering negotiation and step-up authentication
 - Changed the PAT and AAT OAuth scopes to be plain strings

UMA demonstrations and discussions: patient-centric health data sharing



Next steps

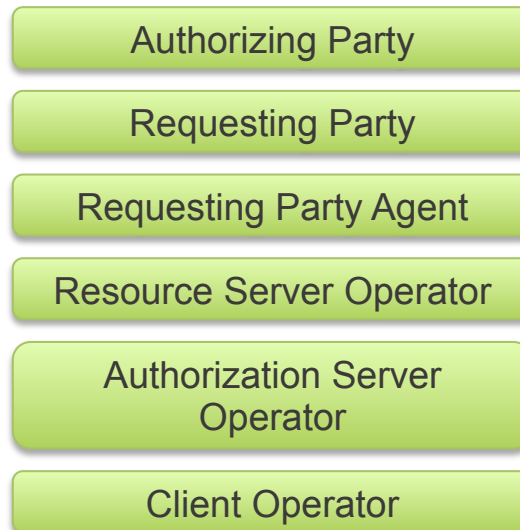
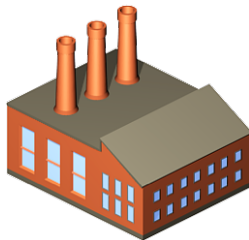
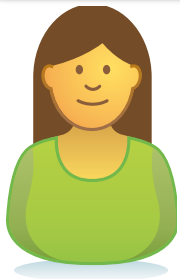
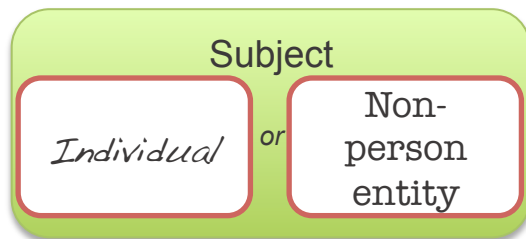


Technical efforts

- Currently: Implementations, deployments, and V1.0 errata-gathering, V.next issue-gathering
- Imminently: Gear up on funded test suite development and “testing the test suite”
- If called for, spec revisions for errata in Q3
- Target end-of-year “Roland testing”
- Intend to do IETF Independent Submissions as Informational RFCs

UMA Binding Obligations

- Distributed authorization across domains? Scary!
- This draft spec contains legalese so parties operating and using software entities (and devices) can distribute rights and obligations fairly
- Trust frameworks = *Access federations*
- Opportunities for liaisons with Kantara Consent Receipts, OTTO*, VOT†...



New obligations (and rights) tend to appear at important protocol “state changes”

*Open Trust Taxonomy for OAuth2

†www.ietf.org/mailman/listinfo/vot

Thanks to all the UManitarians!



Questions? Thank you!

Eve Maler, chair (@xmlgrrl)

Maciej Machulak, vice-chair (@mmachulak)

@UMAWG | tinyurl.com/umawg

16 May 2015

