

User-Managed Access

UMA Work Group

@UMAWG

tinyurl.com/umawg | tinyurl.com/umafaq

29 Oct 2013

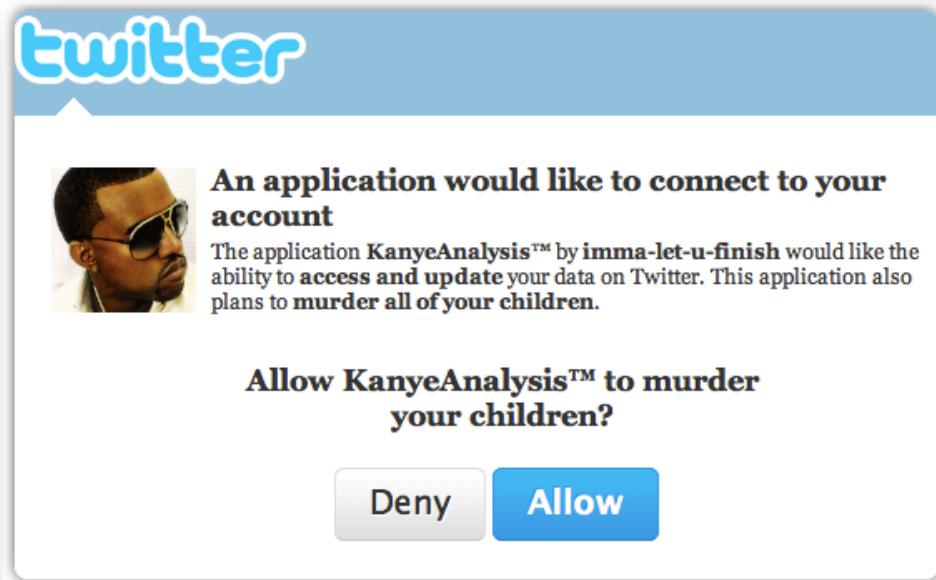


The “data price” for online service is too high: typing...

- Provisioning by hand
- Provisioning by value
- Oversharing
- Lying!

| | |
|-------------------------|--|
| Name | <input type="text"/> |
| Street Address | <input type="text"/> <input type="text"/> |
| City | <input type="text"/> |
| State | Enter Text <input type="button" value="v"/> |
| Zip/Postal | <input type="text"/> <input type="text"/> |
| Province | <input type="text"/> |
| Country | Enter Text <input type="button" value="v"/> |
| Phone | <input type="text"/> |
| Email | <input type="text"/> |
| Preferred Communication | <input type="radio"/> Postal Mail <input type="radio"/> Phone <input type="radio"/> E-mail |

The “data price” for online service is too high: connecting...



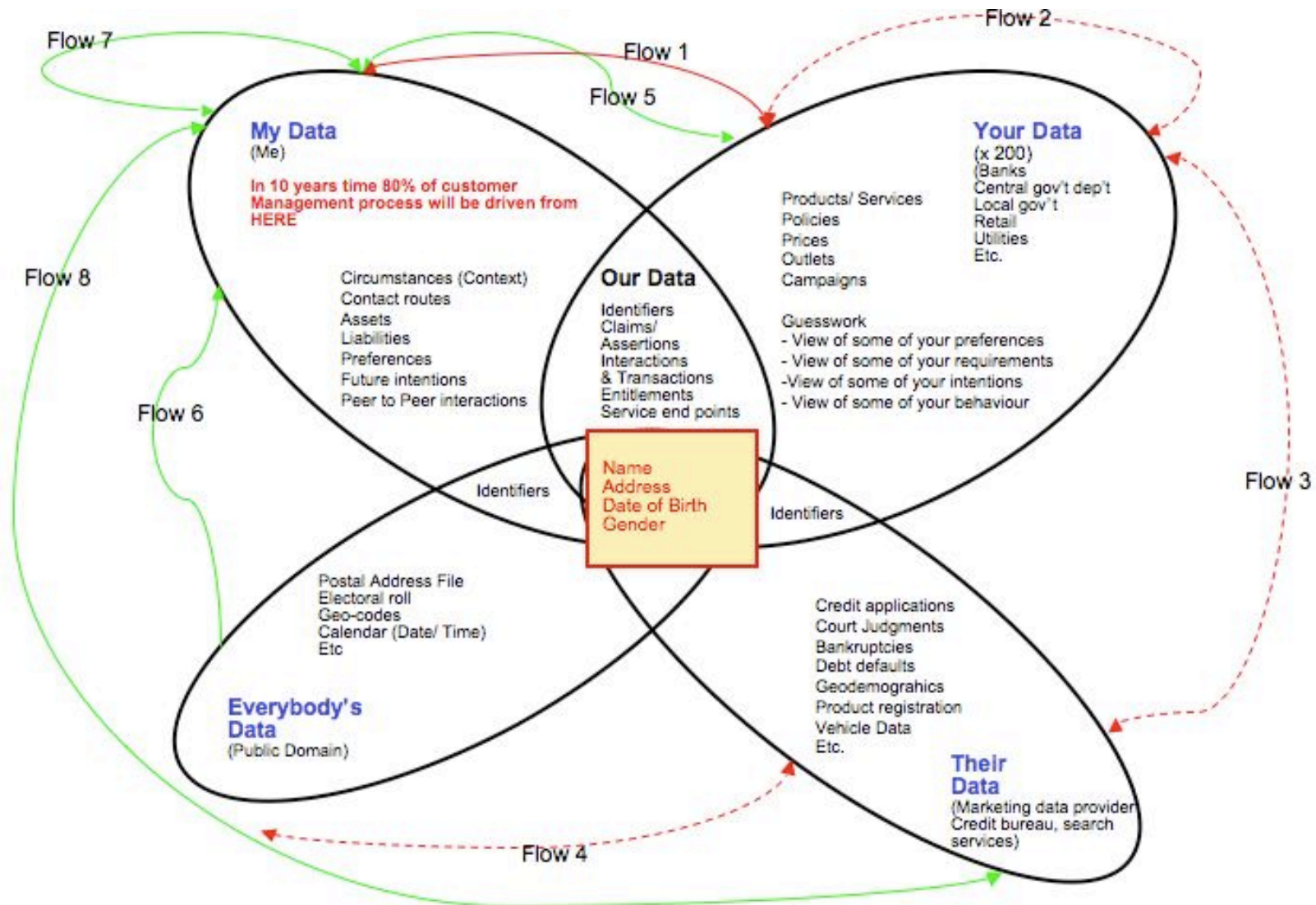
- Meaningless consent to unfavorable terms
- Painful, inconsistent, and messy access management
- Oblivious oversharing

The “data price” for online service is too high: private URLs...



- Handy but insecure
- Unsuitable for really sensitive data

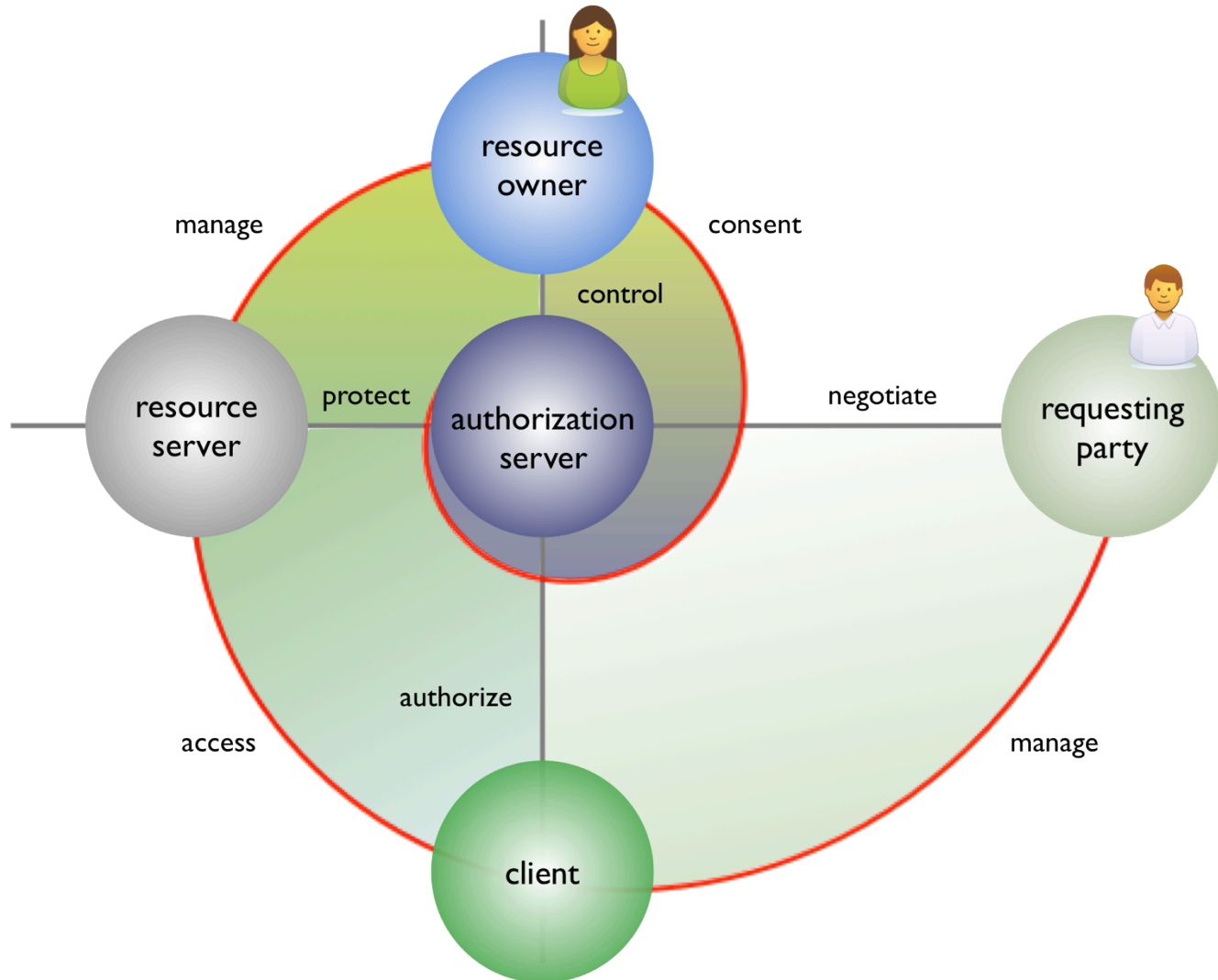
Most data “sharing” today is back-channel and unconsented



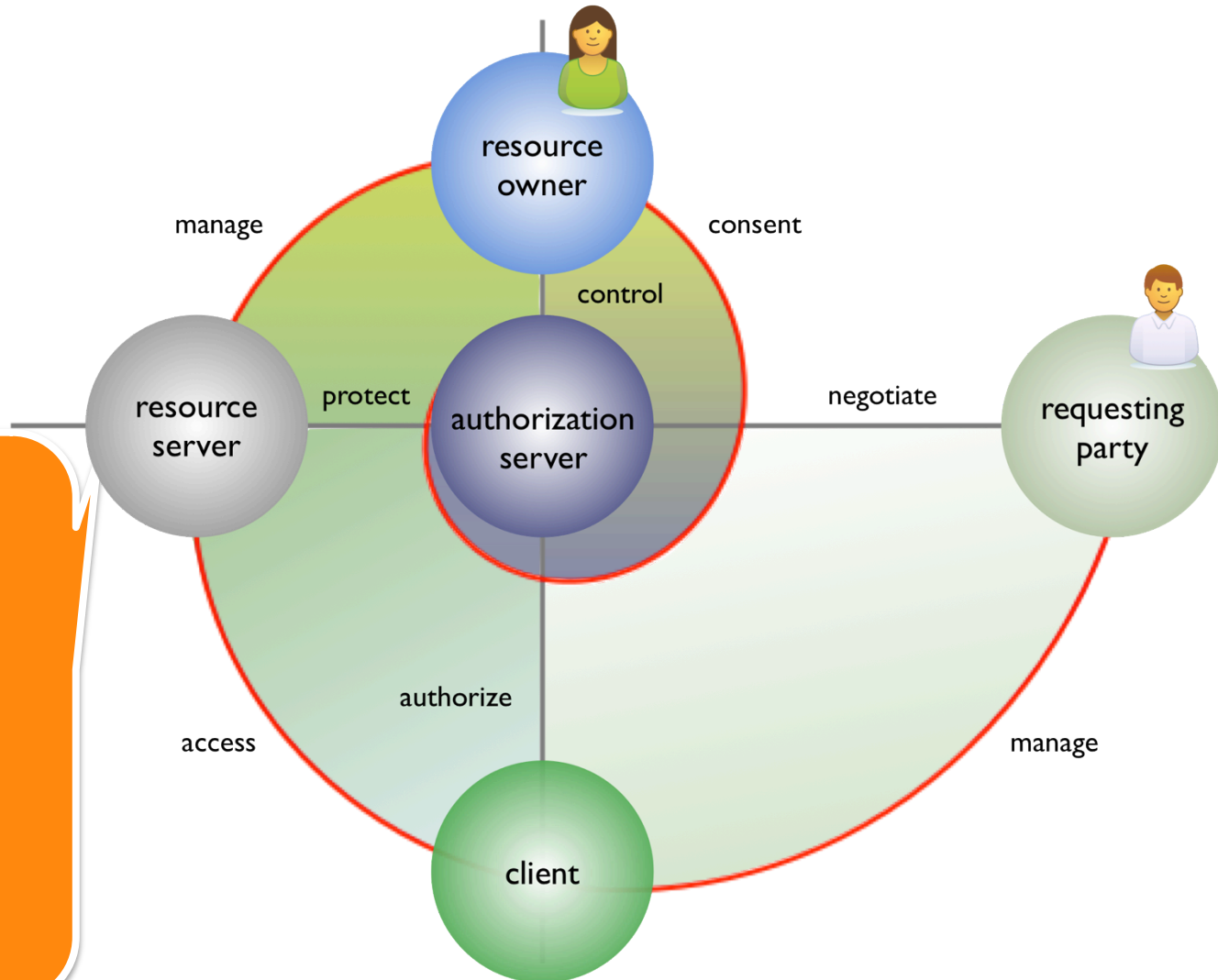
Privacy is about context, control, choice and respect – so UMA enables a “digital footprint control console”

- *Web 2.0 access control is inconsistent and unsophisticated*
- *To share with others, you have to list them literally*
- *You have to keep rebuilding your “circles” in new apps*
- *You can’t advertise content without giving it away*
- *You can’t get a global view of who accessed what*
- You can **unify** access control under a single app
- Your access policies can test for **claims** like “over 18”
- You can **reuse** the same policies with multiple sites
- You can control access to stuff with **public** URLs
- You can manage and **revoke** access from one place

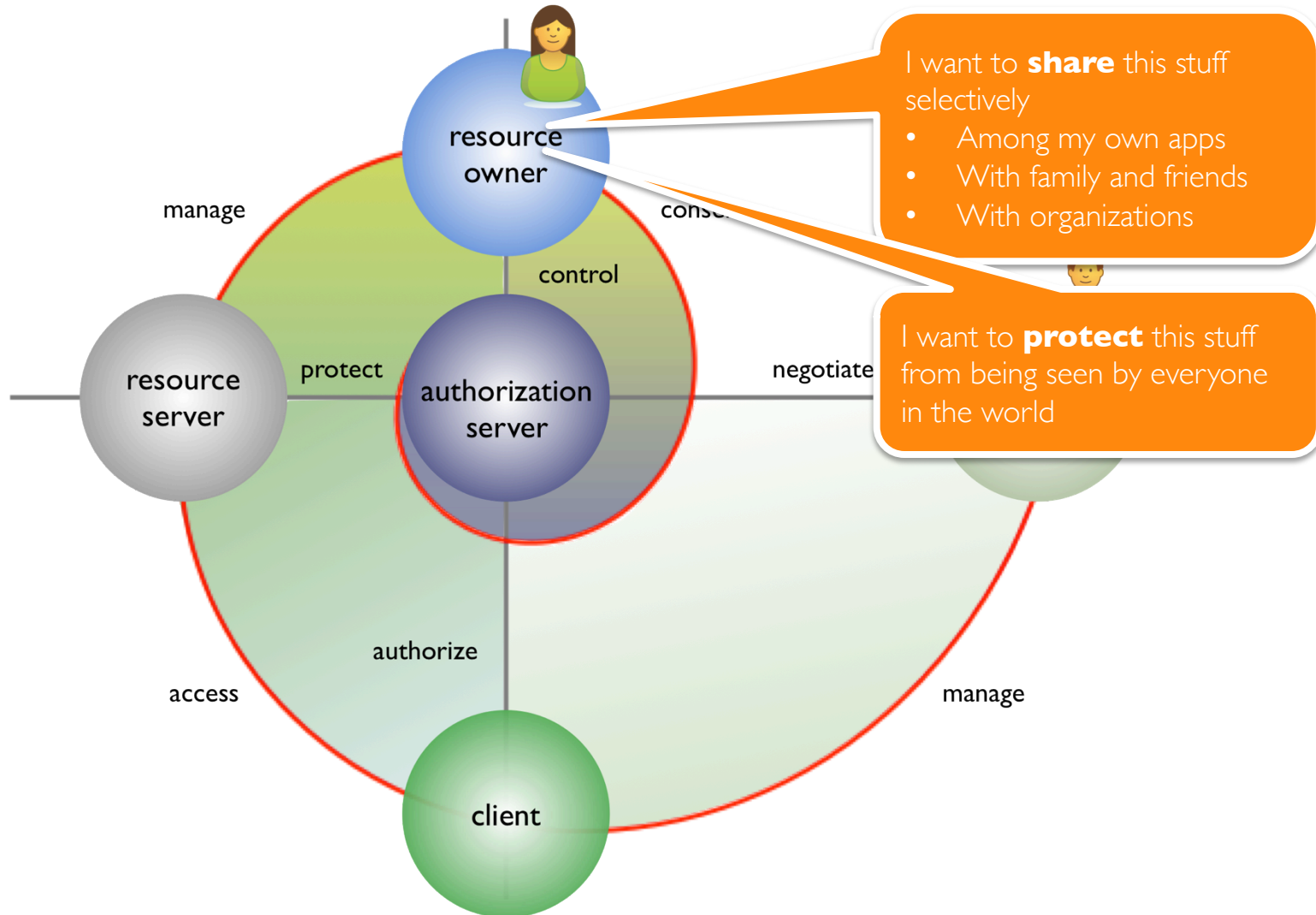
UMA turns online sharing into a privacy-by-design solution



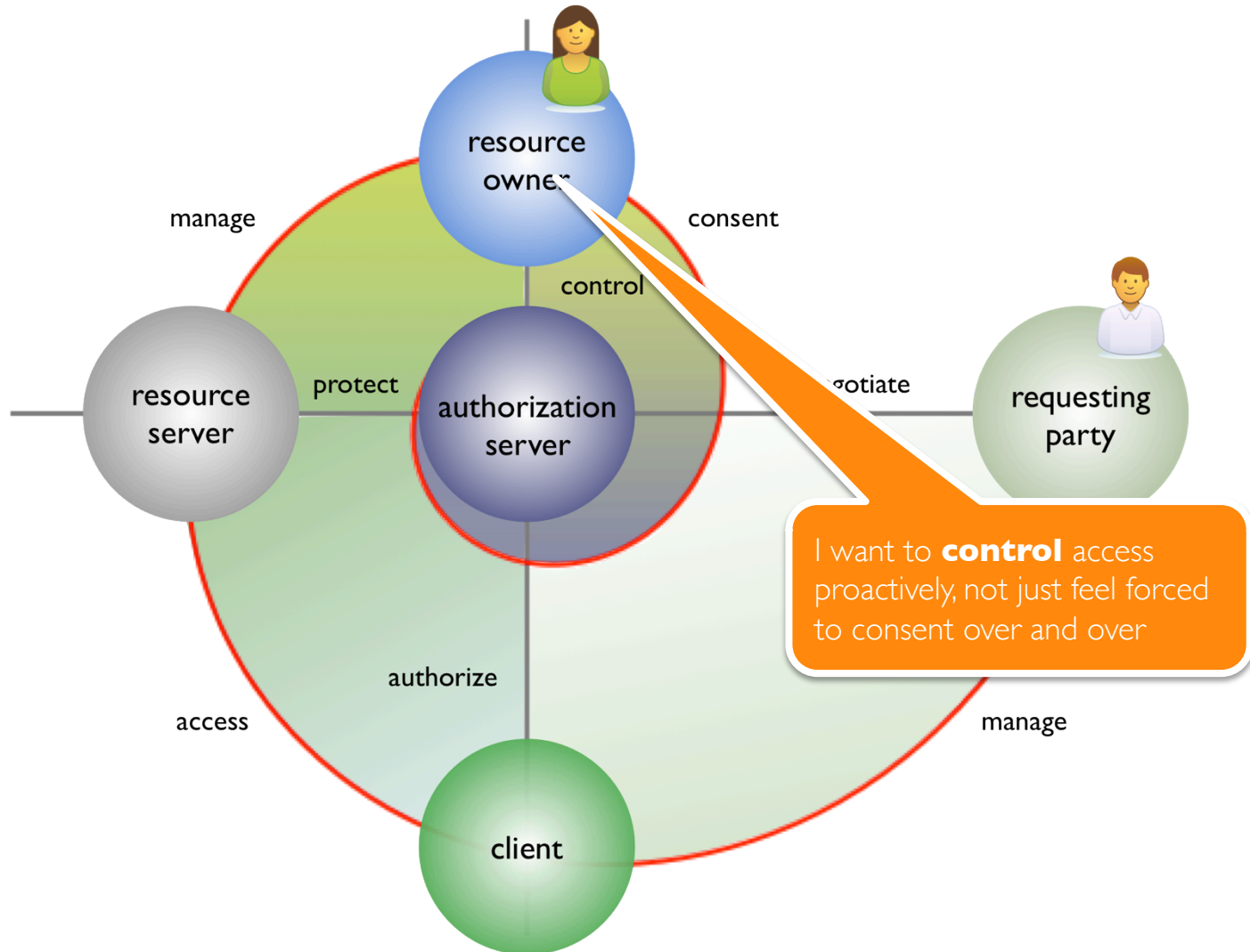
UMA turns online sharing into a privacy-by-design solution



UMA turns online sharing into a privacy-by-design solution



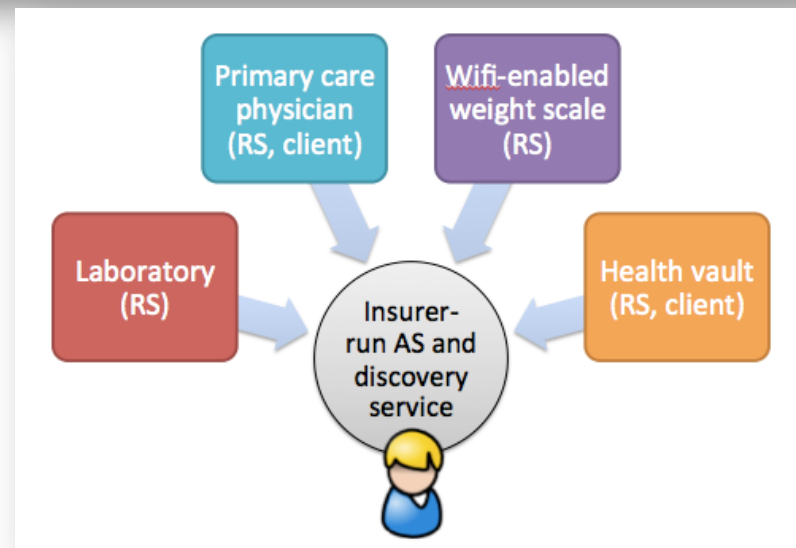
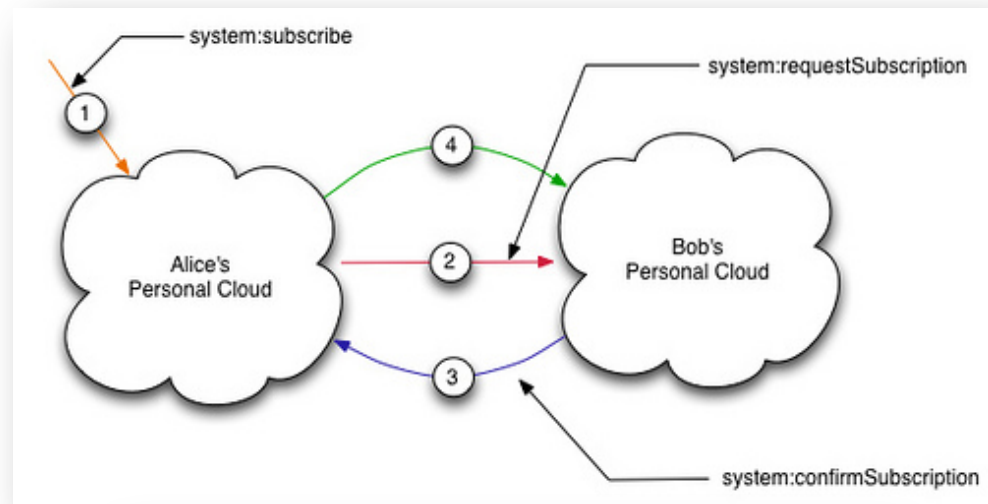
UMA turns online sharing into a privacy-by-design solution



Key use cases

<http://kantarainitiative.org/confluence/display/uma/Case+Studies>

- Subscribing to a friend's personal cloud
- Sharing accessibility attributes ("GPII")
- E-transcript sharing ("HEAR")
- Patient-centric health data access
- Enterprise "access management 2.0"

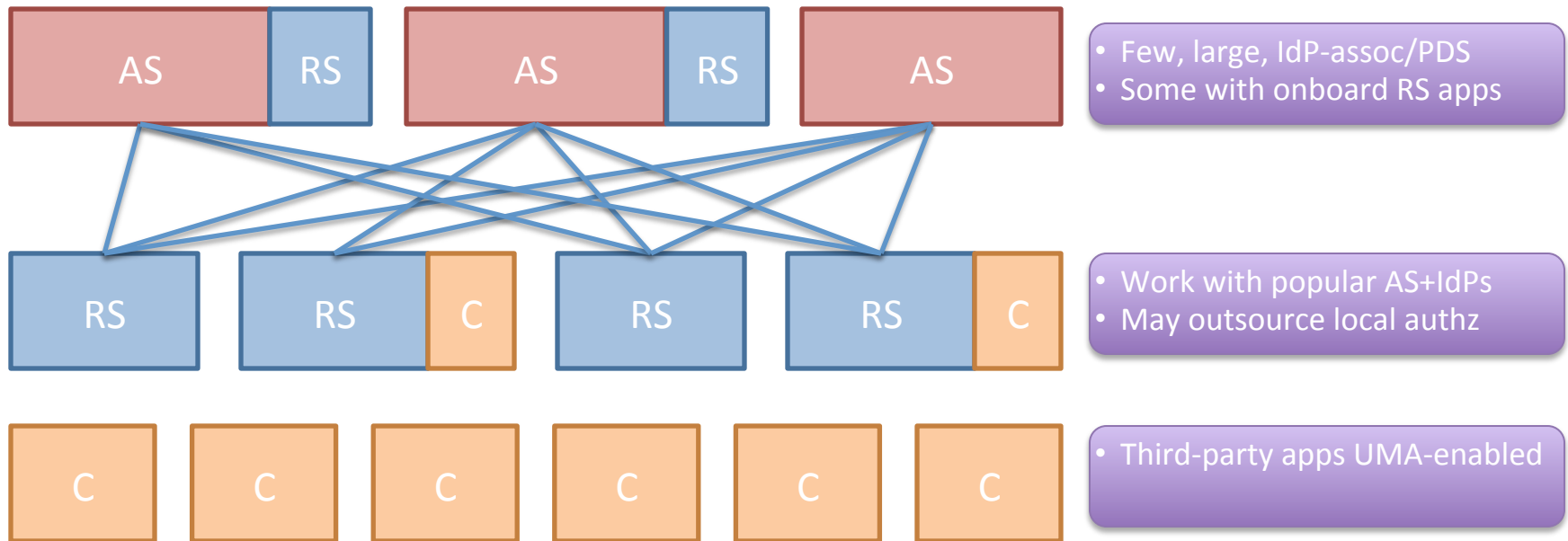


Enterprise use cases bring WAM into the API economy

- *Scopes are entirely proprietary and non-interoperable*
- *Access management and policies are done on a pairwise, per-service basis*
- You create and standardize machine-readable scope descriptions
- You can centralize scope mgmt at one AS and reuse policies
- The RO is the enterprise itself
- The policy administrator is an “RO agent”
- The AS is a PAP and (pseudo) PDP that can serve as a PIP client

Potential ecosystem: “social access control” (à la social sign-in)

Most dynamic; Alice-to-Bob sharing is the key differentiator



Benefits

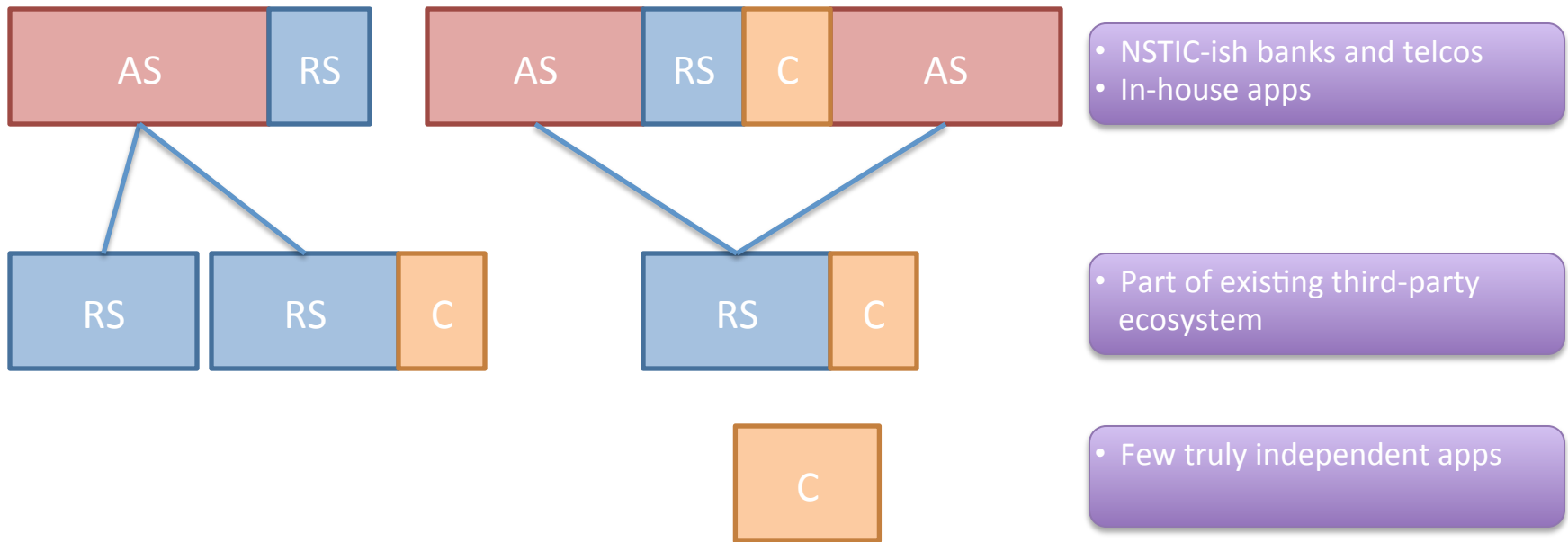
- High-quality, centralized consumer authz

Challenges

- Disruptive change to biz models
- Trust and assurance
- API interoperability

Potential ecosystem: “walled garden PDS’s”

Likely highly static partnerships; Alice-to-Alice/Bob/org sharing



Benefits

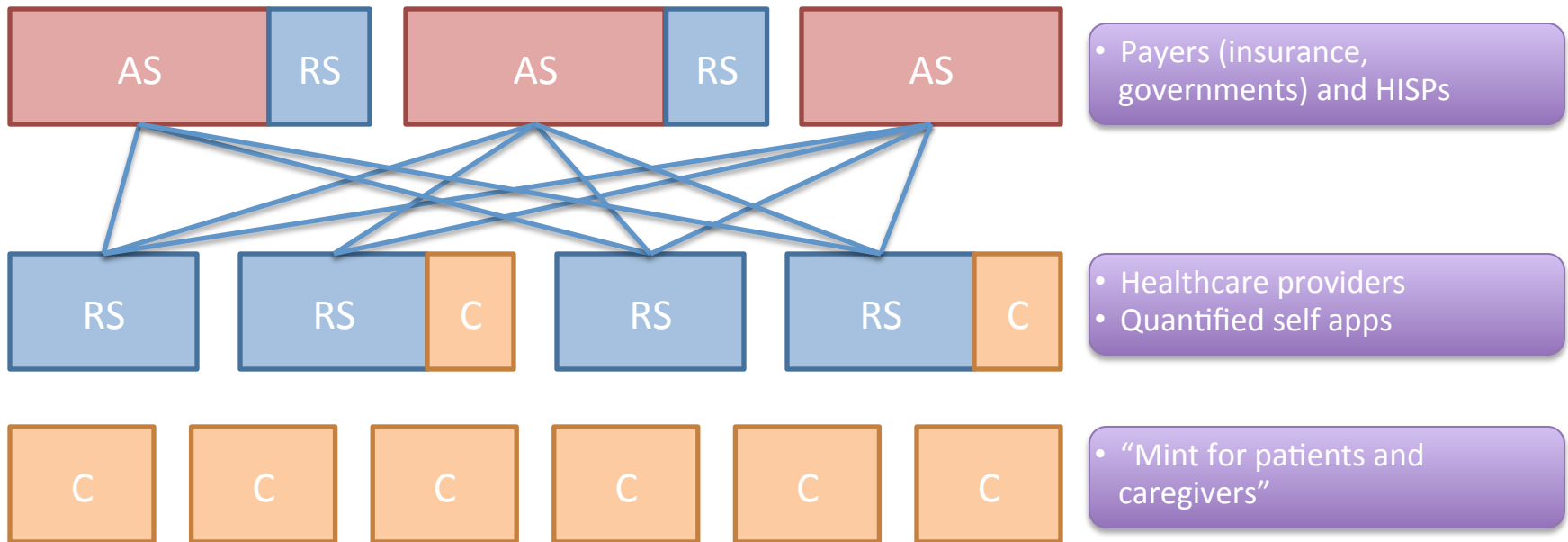
- Today’s back-channel user data is put under user control/monitoring
- “Outward” trust growth

Challenges

- Tight binding to the owner of the garden

Potential ecosystem: “patient-centric health vaults”

Static partnering will center on payers as 900-lb gorillas; highly vertical



Benefits

- Proactive, trackable consent directives
- Blue Button-like delivery of data

Challenges

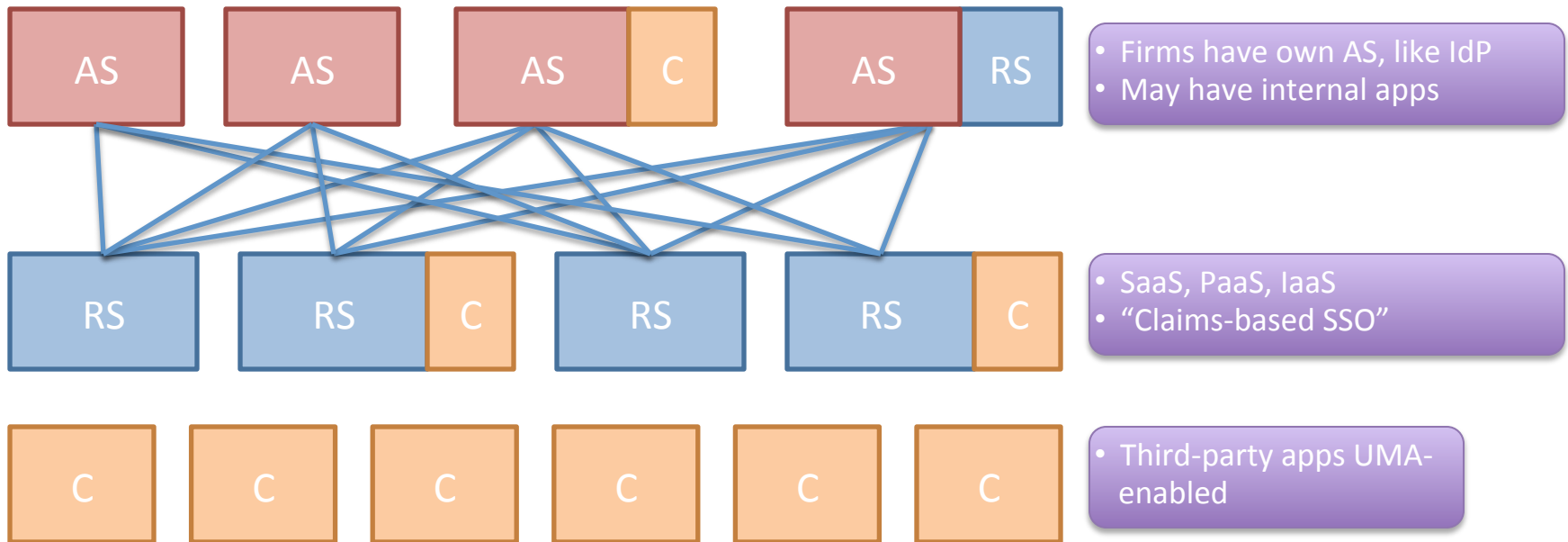
- Sclerotic IT practices
- Serious security, privacy, and discoverability needs

Protocol vs. value-add: the basics

- Apps can outsource reusable high-quality access control
- Your access policies can test for claims like “over 18”
- You can delegate constrained access to autonomous others
- You can control access to stuff with public URLs
- You can manage and revoke access from one place
- You create and standardize machine-readable scope descriptions
- You can centralize scope mgmt at one AS and reuse policies
- Protocol + likely AS/RS agreements
- Protocol + policy/claim support in AS UX and functionality
- Protocol + policy/claim support in AS UX and functionality
- Protocol + “personal discovery” features
- AS UX and functionality
- Profiling
- Protocol

Potential ecosystem: “distributed authz for business” (access management 2.0)

AliceCo-to-Employee/Contractor/PartnerBob sharing



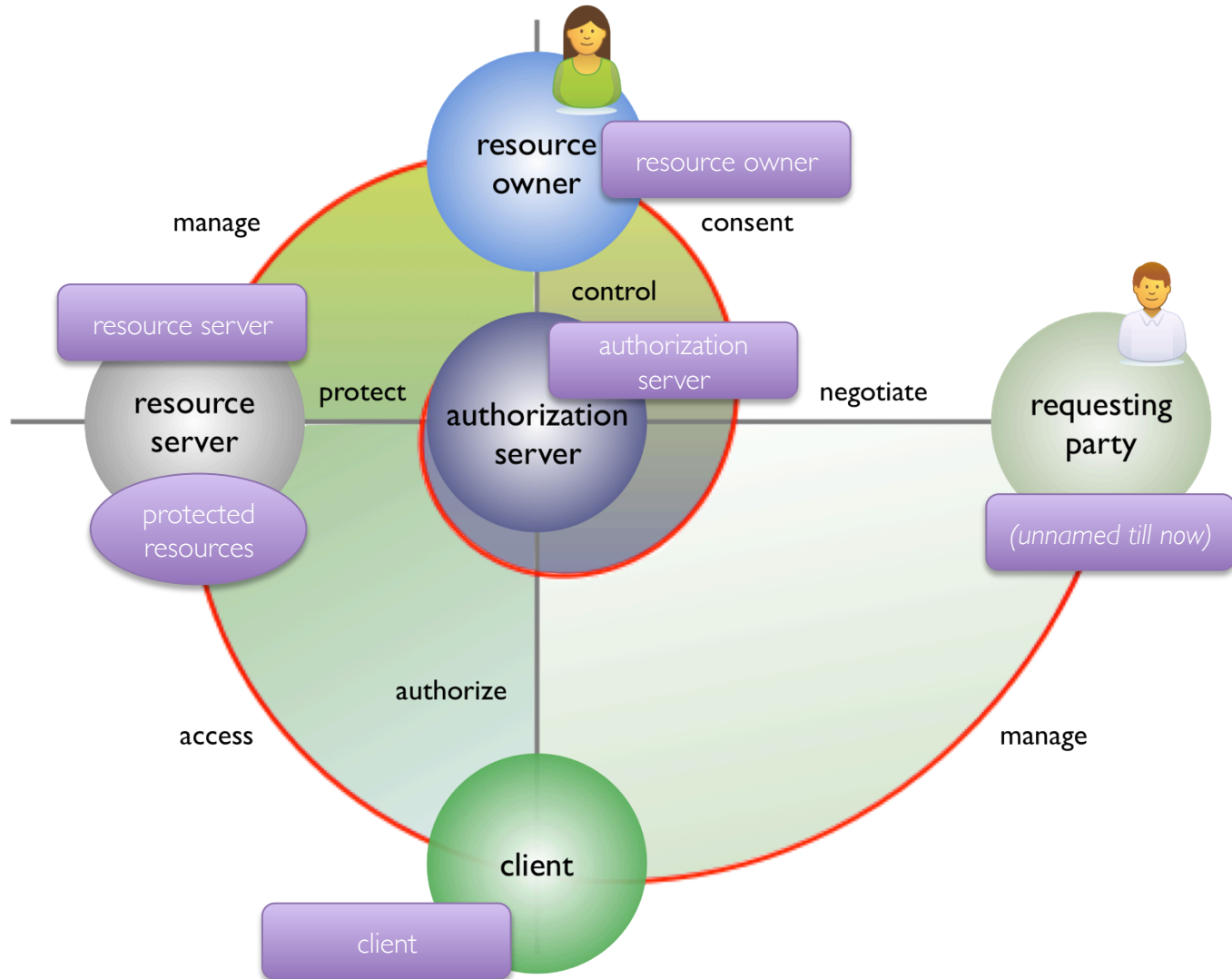
Benefits

- Centralized scope mgmt across web, mobile
- Less dependent on a “big bang”

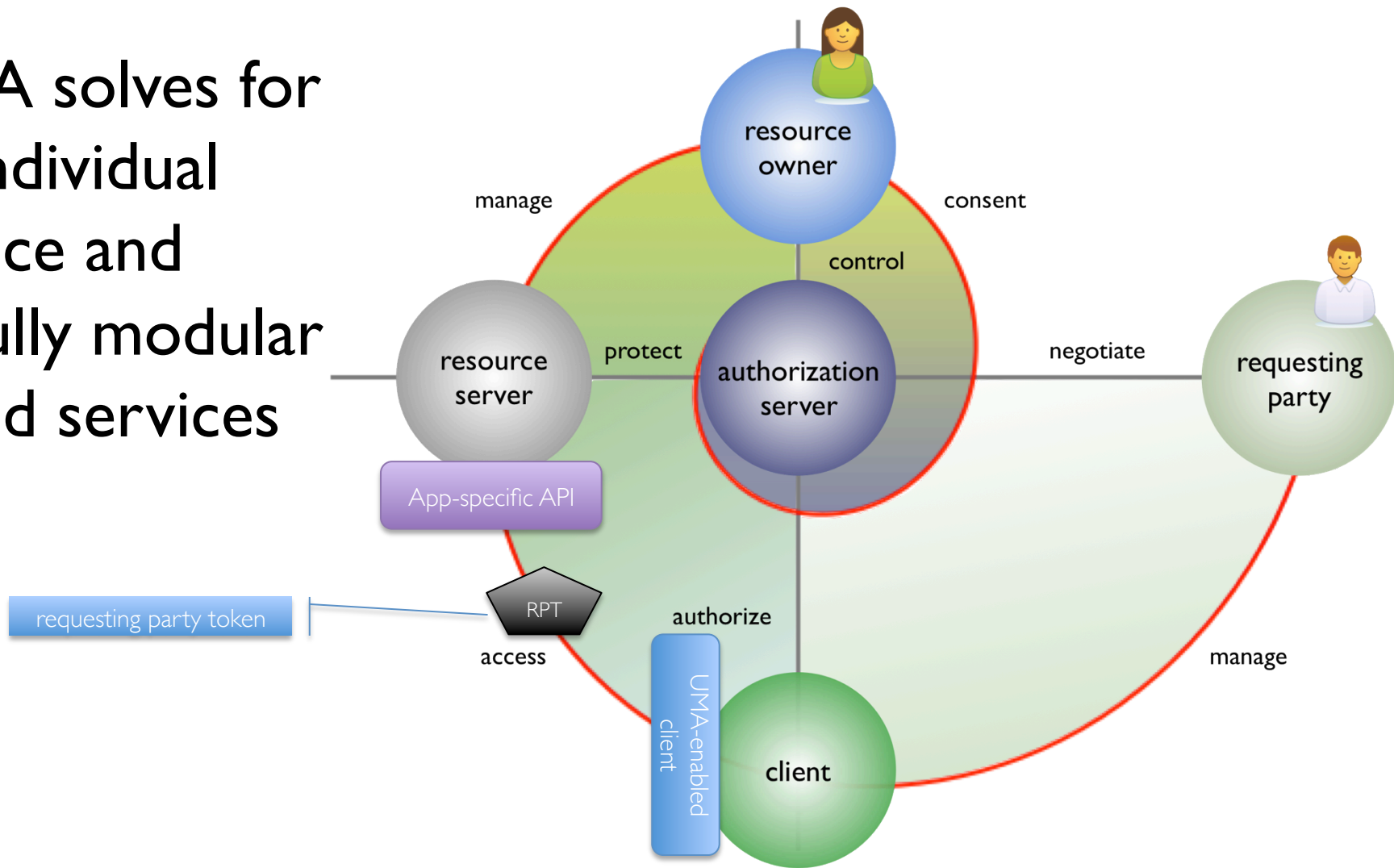
Challenges

- Legacy apps and WAM practices

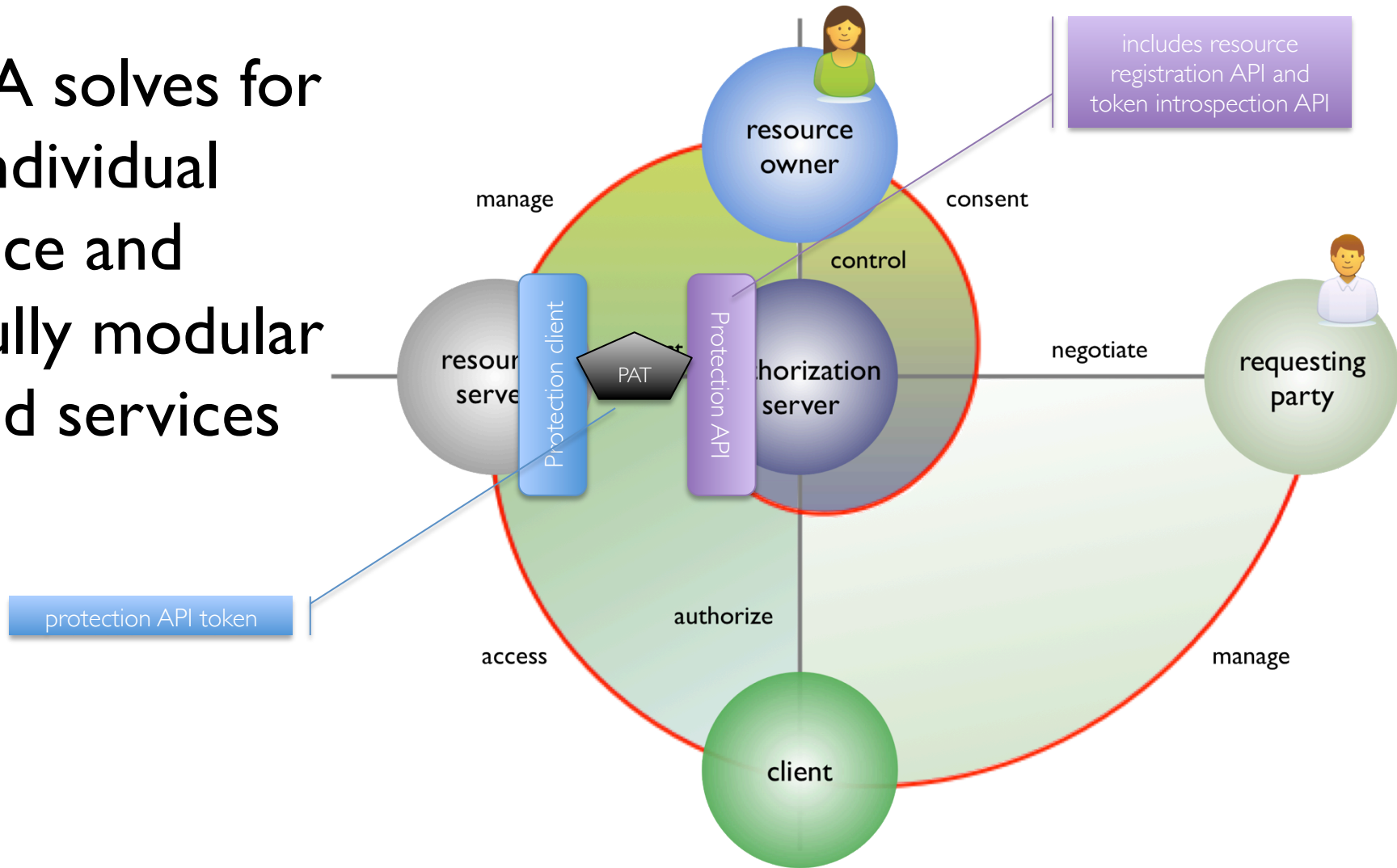
UMA is a profile of OAuth, with bits added for interop and scale



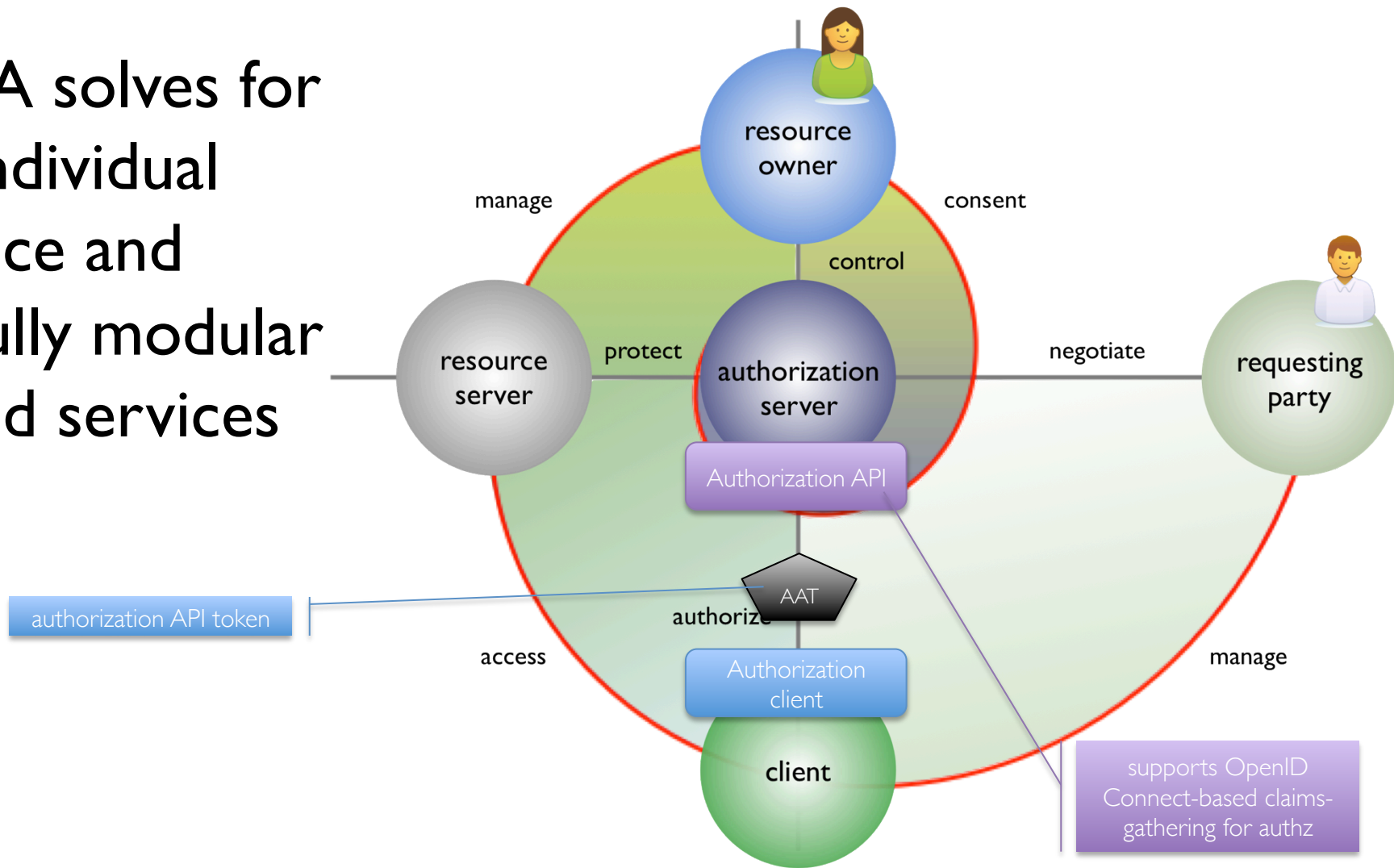
UMA solves for
1) individual
choice and
2) fully modular
cloud services



UMA solves for
1) individual
choice and
2) fully modular
cloud services



UMA solves for
1) individual
choice and
2) fully modular
cloud services



Key implementations

<http://kantarainitiative.org/confluence/display/uma/UMA+Implementations>

- SMARTAM.net (running authorization service from Cloud Identity UK)
- Puma (Python libraries for RS- and client-enabling web apps) from ditto
- Fraunhofer AISEC open-source implementation in Java
- Gluu OX open-source implementation for Access Management 2.0 use cases



Next steps

- UMA has several independent implementations, some available as open source
- UMA interop activities are ongoing
- Work is under way with legal experts on “access federation” trust frameworks
- Case studies, FAQ, and more are available
- Get involved!
 - Follow @UMAWG
 - Become an UMANitarian (it’s free!)
 - Join the UMA-dev mailing list
- Visit tinyurl.com/umawg for all the info you need

Questions? Thank you

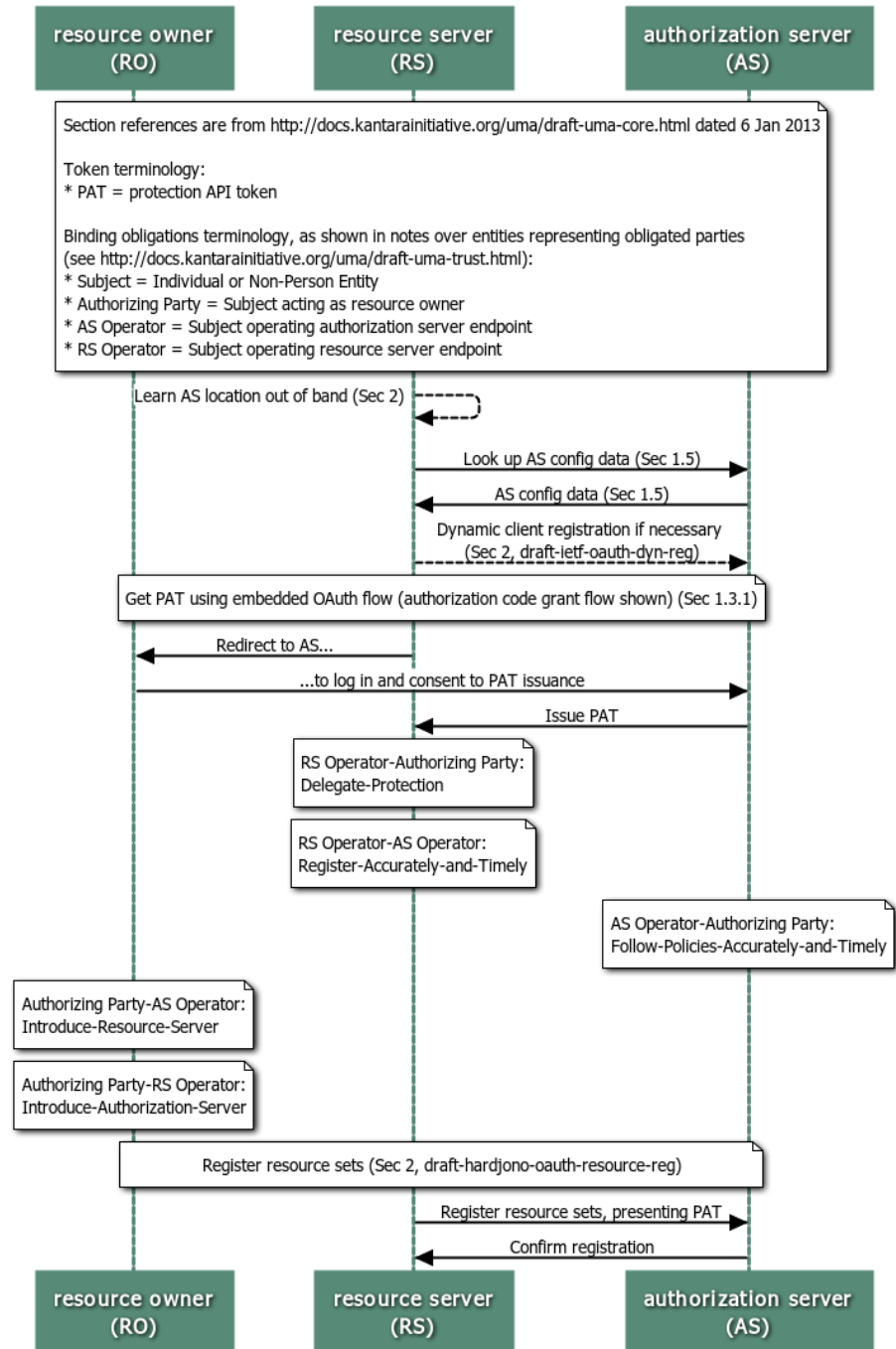
@UMAWG

tinyurl.com/umawg | tinyurl.com/umafaq



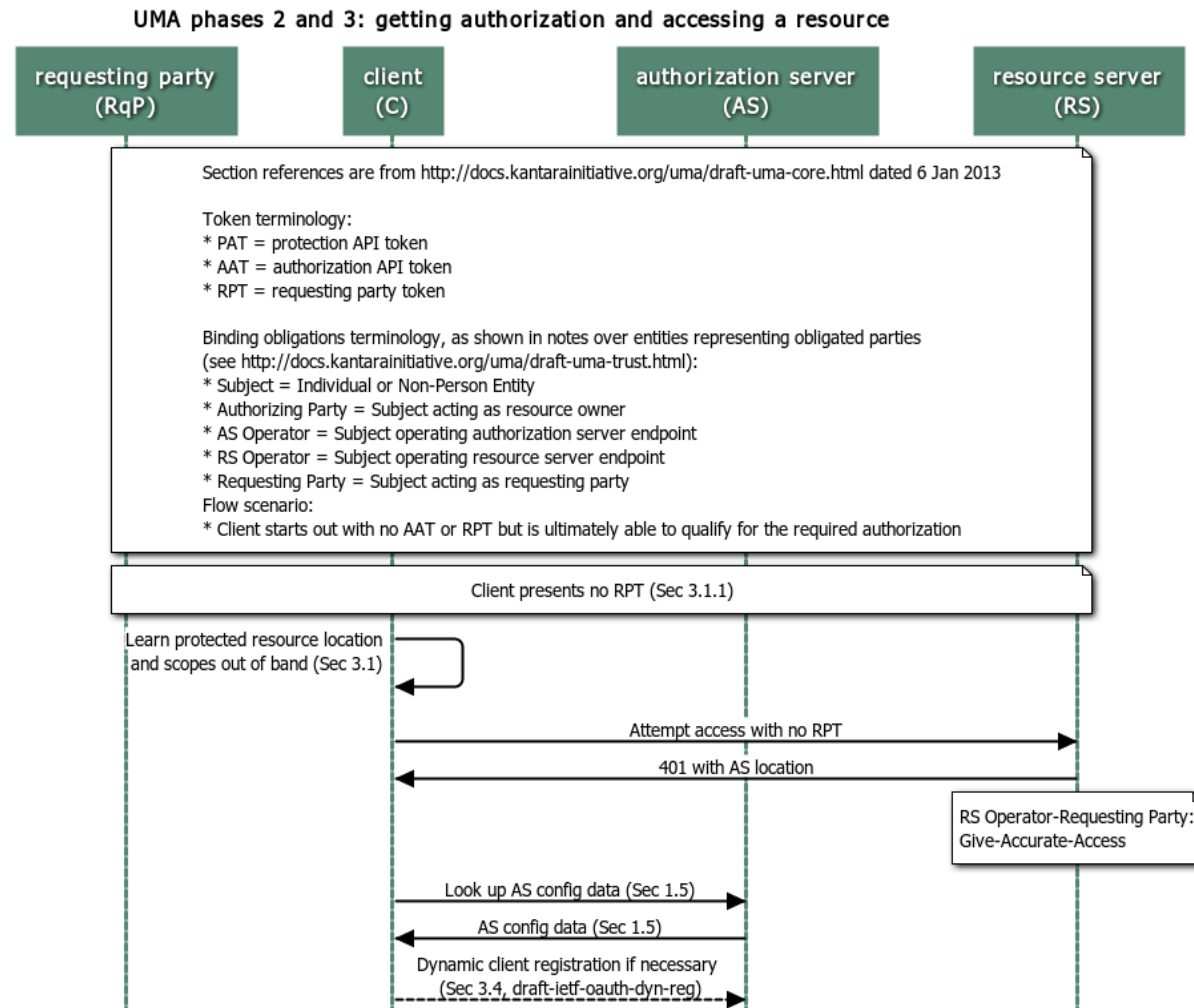
Phase I: protect a resource

UMA phase 1: protecting a resource (rev 07b)



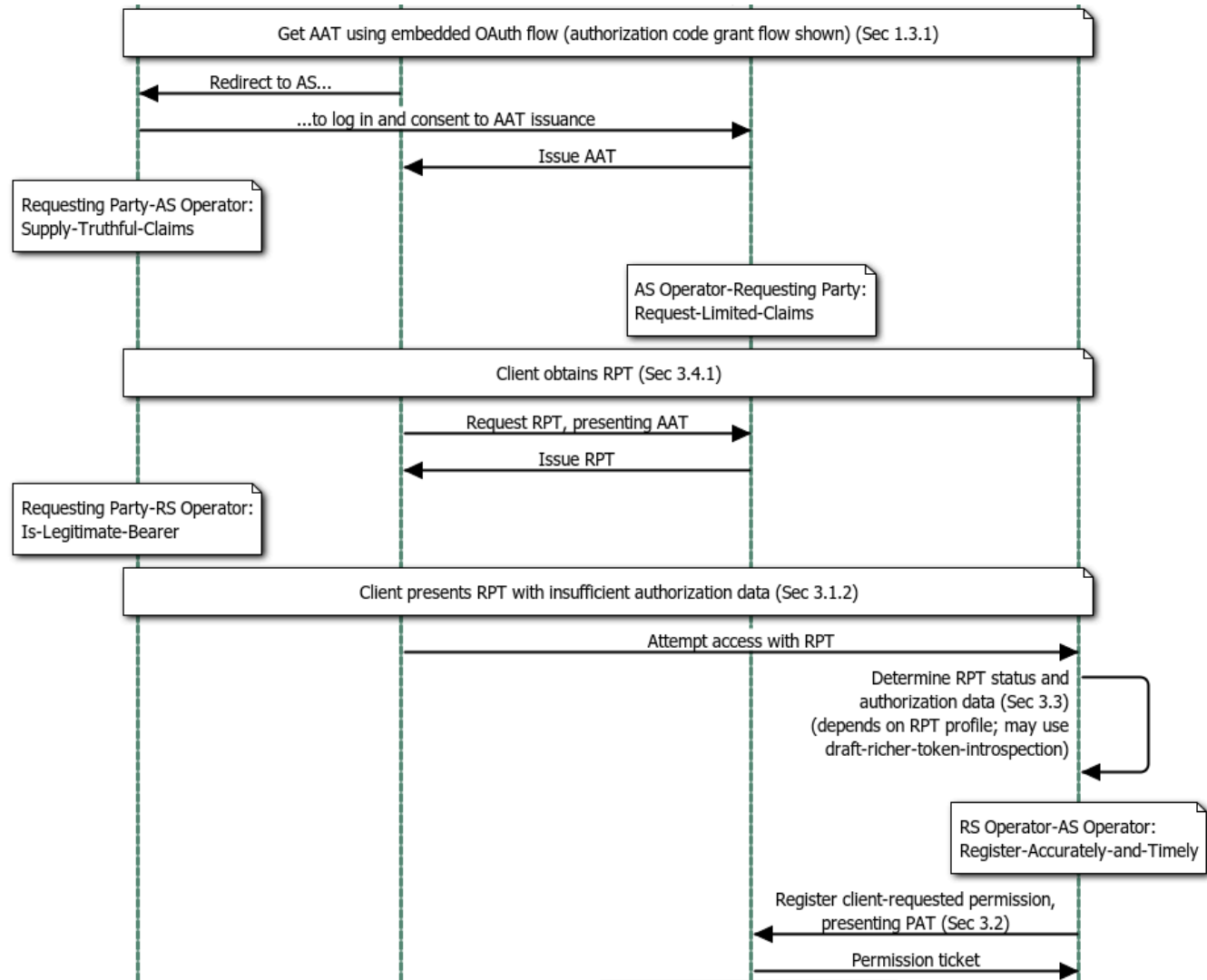
Phases 2 and 3: get authorization and access resource

1 of 3



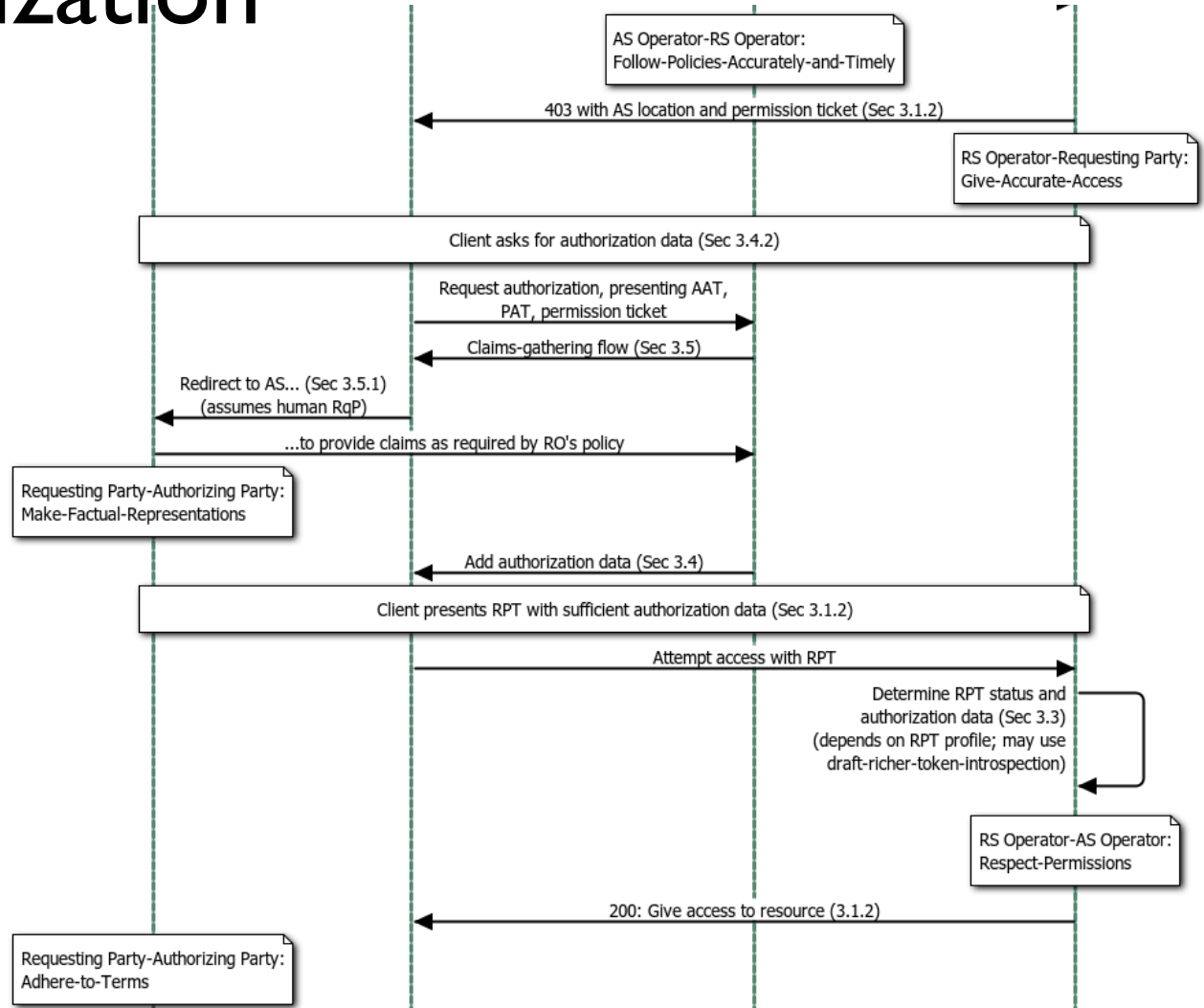
Phases 2 and 3: get authorization and access resource

2 of 3



Phases 2 and 3: get authorization and access resource

1 of 3



Spec call tree for the UMA profile of OAuth

