FORGEROCK™

# Custodianship in and around UMA

Eve Maler
eve.maler@forgerock.com

March 9, 2015

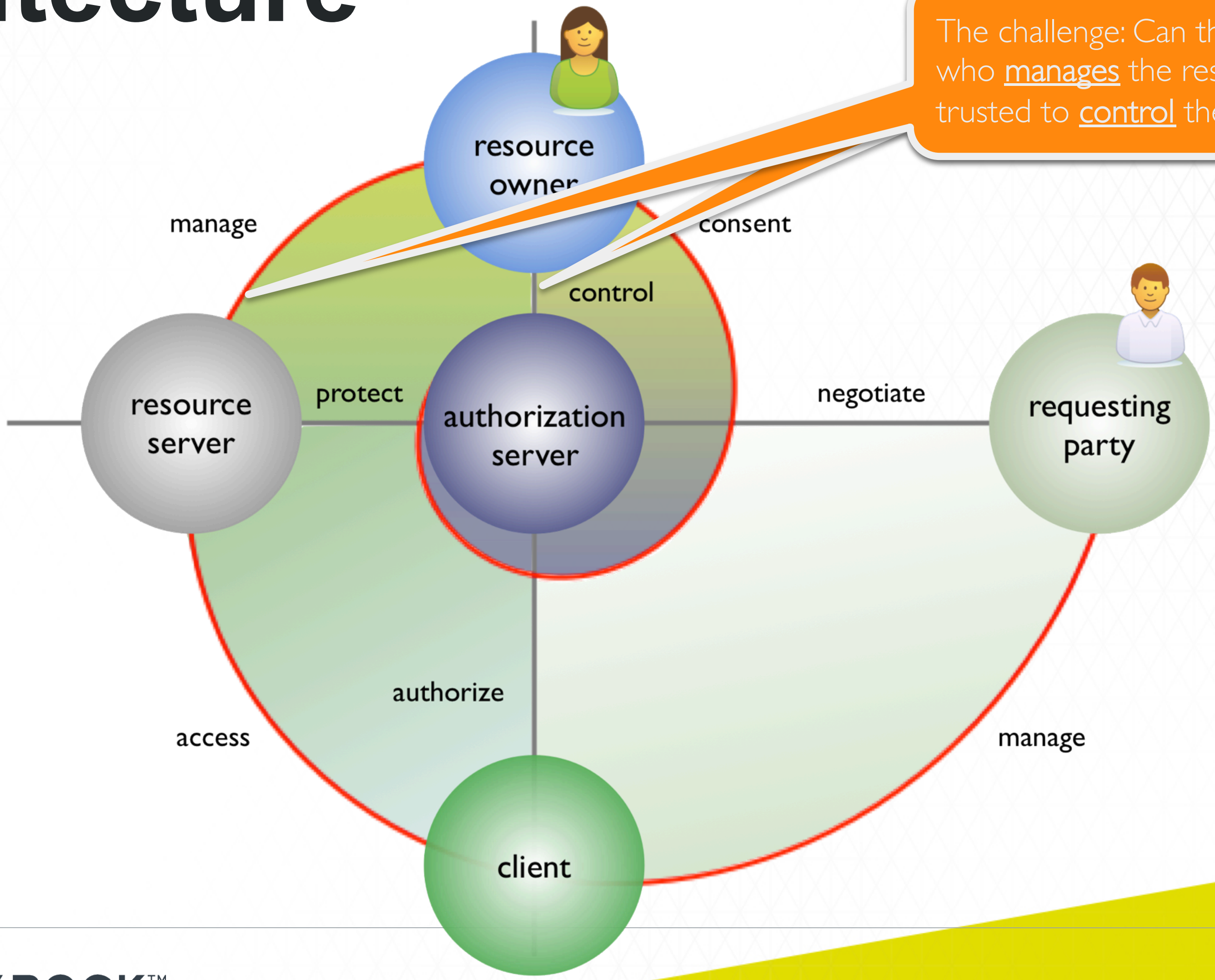
## Examples

- Under-13 student using a school portal to share completed homework assignments
- Elderly parent with intermittent dementia using social networking and health-related apps
- Developmentally disabled adult with access to online bank accounts and related data

## Goal

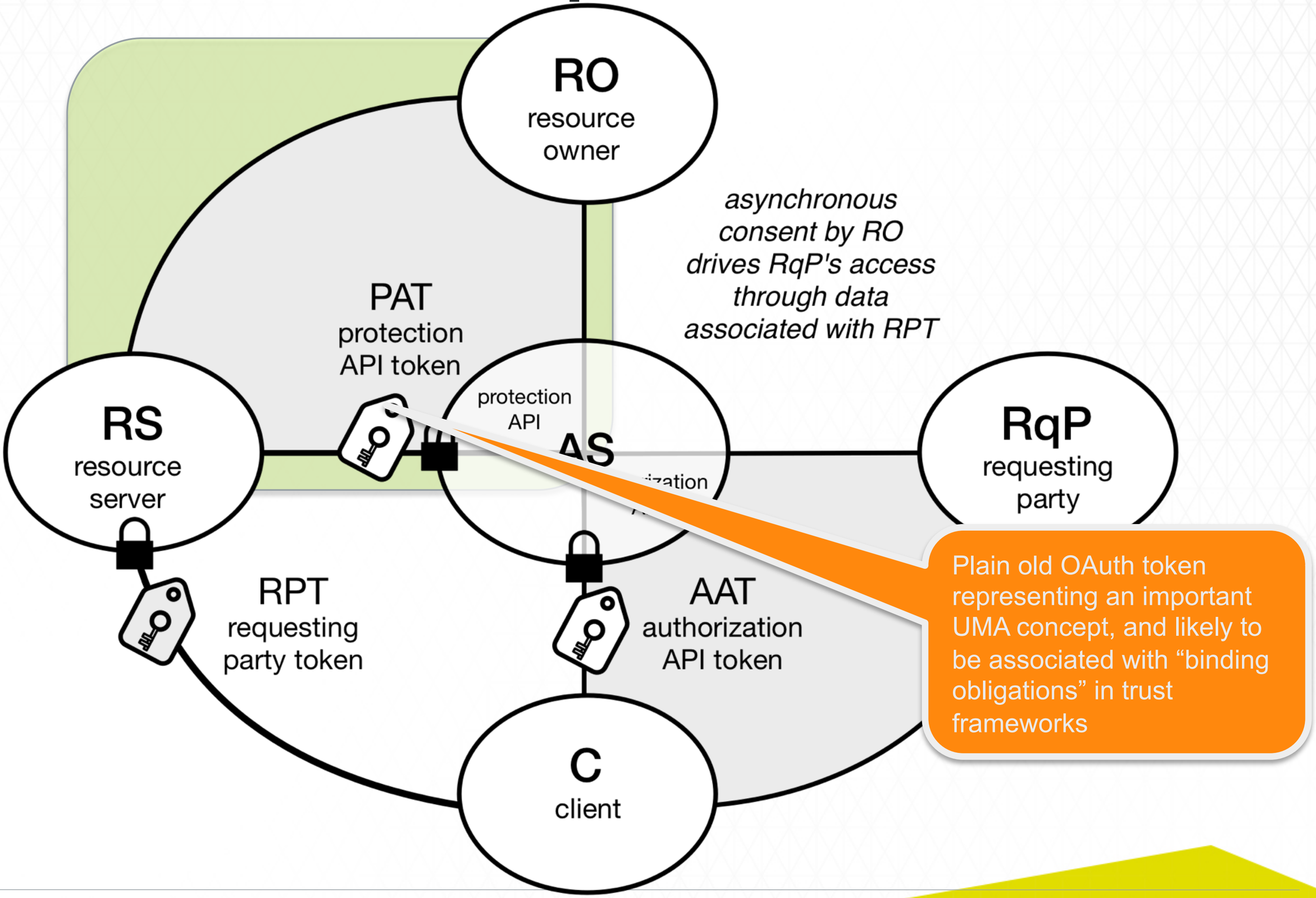
- Bring various benefits of *user-managed access* even to those who are at some level “incapacitated” with respect to consent, if possible

# Reminder about UMA architecture



The challenge: Can the person who manages the resources be trusted to control their access?

# A "PAT" represents the RO's consent for the RS to outsource protection to the AS



Plain old OAuth token representing an important UMA concept, and likely to be associated with "binding obligations" in trust frameworks

FORGEROCK™

# Notes

- UMA does not have a formal notion of "multiple ROs" (like "joint bank accounts")
  - Rationale: V1.0 speed and simplicity; inherent complexity in clashing policy between ROs; Google Apps as existence proof of successful single-RO model
  - Mitigating this "lack" at the app level: If an API exposes wide-ranging **admin** or further-downstream **share** scopes, then an RO can grant them
- As an aside, the RS and C may, in fact, be instances of the same app

# Generic roles used in the following discussion

- Guardian (custodial role)
- Ward (in custody)
- Agent (representative of relevant public/private sector bureaucracy)

## Some options for handling custodianship in and around UMA

*You may have others in mind…*

# Option 1: offline guardian

**RqP (becoming a downstream RO) account initiated under the control of an "offline" guardian**

1. Guardian executes a paper consent form
2. Agent creates an RO account record on guardian's behalf
3. Agent issues PAT on guardian's behalf
4. System-default policy under trust framework – or agent, manually – issues relevant RqP permissions to associated ward
5. Ward can function normally as a downstream RO; however…
6. Guardian, through agent, can monitor control, and revoke ward's access as an RqP as required

## Option 1 discussion

- This a fairly “top-down” pattern
- The offline/proxy pattern seems to match many current public-sector and financial use cases
- The PAT gives some auditability
- Policies/trust framework force some formal accountability
- The onus is on agents to make the whole thing work
- …

## Option 2: online guardian

**RO account initiated by a ward but in a trust framework bounded by an “online” guardian**

1. Ward registers for an RO account; process requires linking a verified guardian’s account, treated as an automatic RqP
2. System-default policies limit ward’s ability to share with others besides guardian RqP; trust framework ensures that ward can monitor uncontrolled disclosures; standard scopes ensure extent of access by guardian

## Option 2 discussion

- This is a fairly “bottom-up” pattern
- The online pattern seems to be closer to some private-sector use cases
- System-default policies and, particularly, scopes give guardian some real “teeth” for overseeing ward’s activity
- …

# Option 3: “outside UMA”

## Enhanced AS handles RO impersonation duties

- Kennisnet has chosen this option for its LACE Project in the Netherlands for K-12 students, currently in UX mockups:
  - http://www.laceproject.eu/blog/give-students-control-data/
  - http://panelpicker.sxsw.com/vote/32086
- Mark Dobrinic of Kennisnet says:
  - *“We have decided that dealing with custodians is a problem by itself. In our case, this means that we have moved the relationship between child-custodian to the AS completely. This is visualized in the Dashboard(AS) application, by the mother that logs in at the Dashboard(AS), and she can select which one of her children she wants to use the dashboard for. …. So, it has been part of our thinking in the design, but we have isolated it away from UMA and thought that projecting it on the AS would allow us to focus on the rest of the case study.”*

# Option 3 discussion

- Impersonation approaches are what UMA tries to avoid!
- But there hasn't been guidance to date on how to go beyond
- …