User-Managed Access (UMA) Working Group

Eve Maler, WG chair

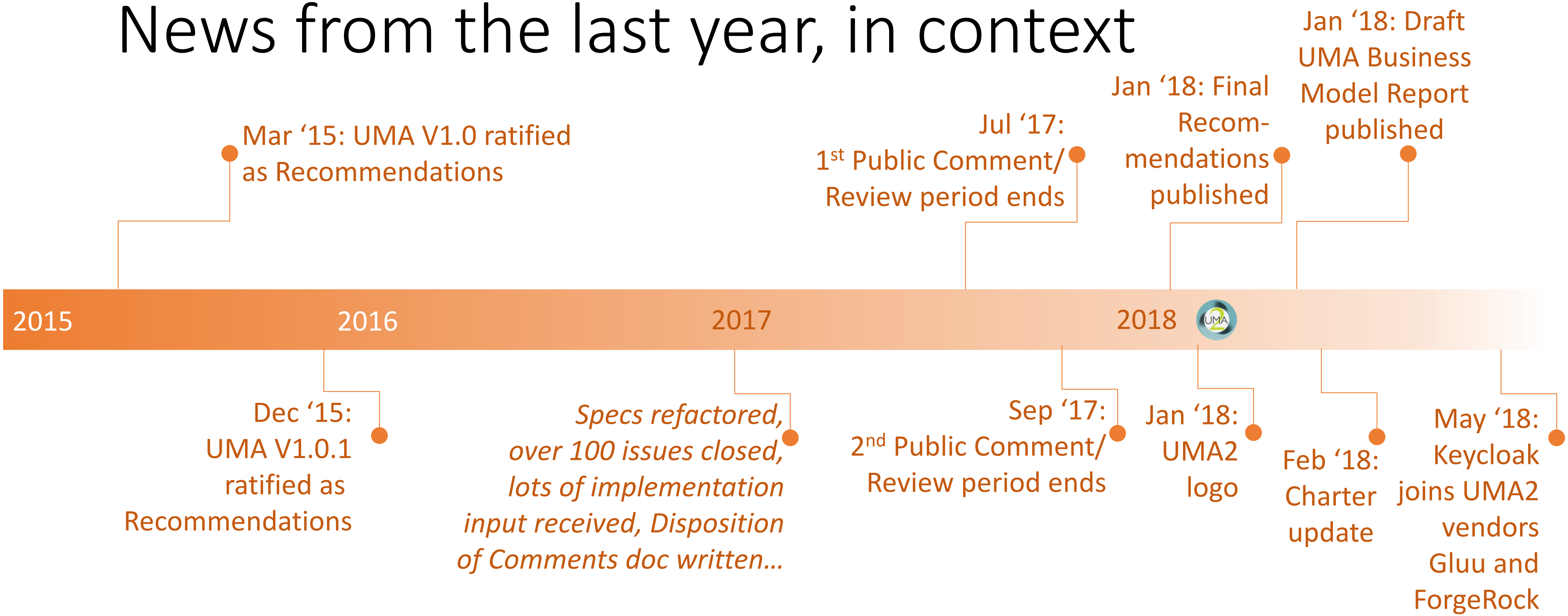
eve.maler@forgerock.com| @xmlgrrl

15 May 2018

http://tinyurl.com/umawg/ | @UMAWG

First: Shall we do a quick UMA explainer? ▶

News from the last year, in context



Mar ‘15: UMA V1.0 ratified as Recommendations

Dec ‘15: UMA V1.0.1 ratified as Recommendations

Specs refactored, over 100 issues closed, lots of implementation input received, Disposition of Comments doc written…

Jul ‘17: 1st Public Comment/ Review period ends

Sep ‘17: 2nd Public Comment/ Review period ends

Jan ‘18: Final Recom-mendations published

Jan ‘18: UMA2 logo

Jan ‘18: Draft UMA Business Model Report published

Feb ‘18: Charter update

May ‘18: Keycloak joins UMA2 vendors Gluu and ForgeRock

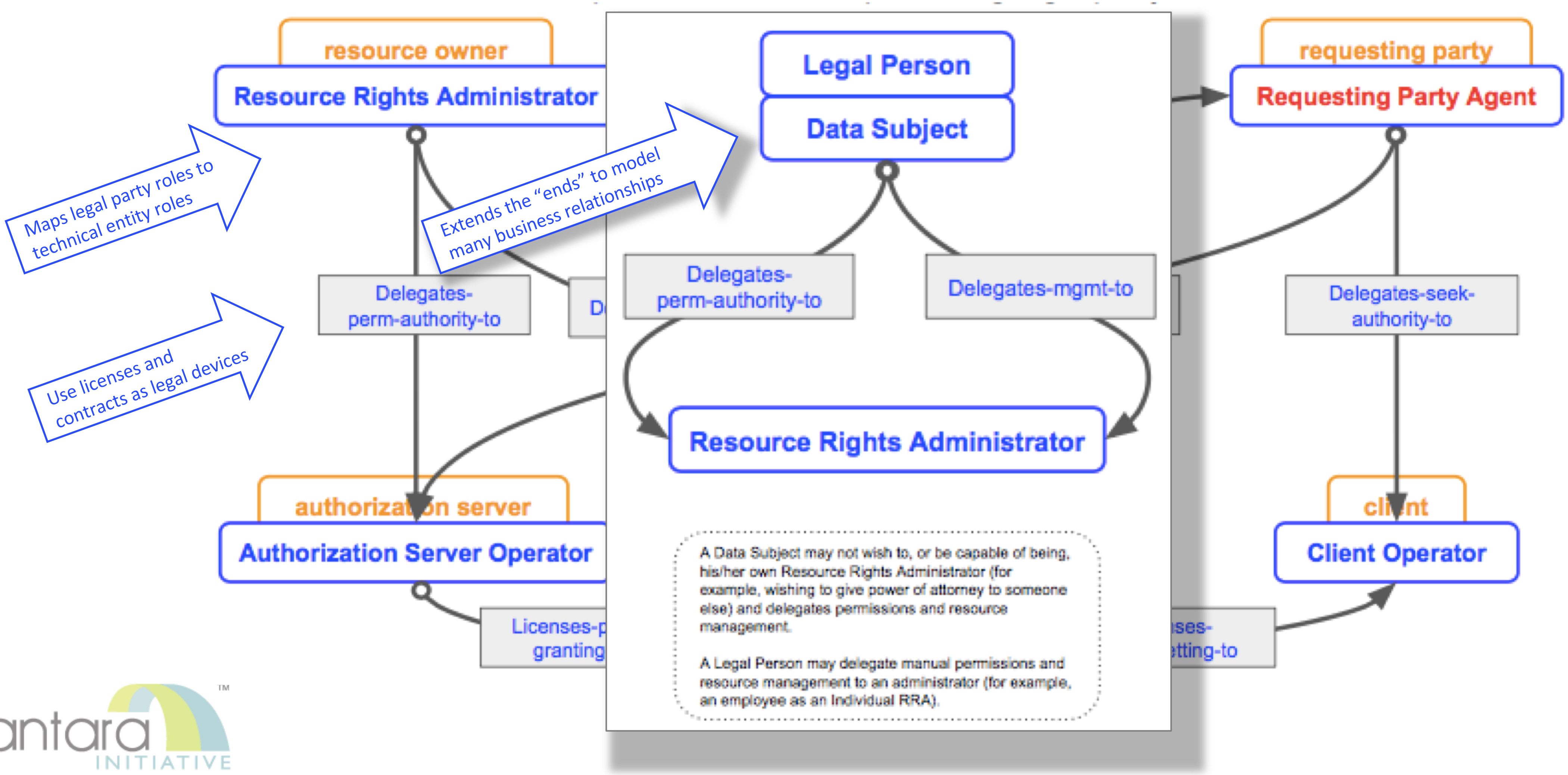
Some use cases/ecosystems involving UMA

- Financial
  - UK Pensions Dashboard Project / OIX / Origo
  - Examining suitability for a set of Open Banking use cases
- IoT
  - “ACE actors” architecture identifies requirements for authorization to an RqP
- Healthcare
  - Profiled in Health Relationship Trust (HEART) at OpenID Foundation
  - Part of the new OpenMedReady framework, along with HEART

ORIGO

OMR
Open Med Ready

The new UMA business model defines how the UMA protocol enables a license-based model for controlling access rights to personal digital assets


resource owner
Resource Rights Administrator
Delegates-perm-authority-to
D
authorization server
Authorization Server Operator
Licenses-p
granting
Legal Person
Data Subject
Delegates-perm-authority-to
Delegates-mgmt-to
Resource Rights Administrator
requesting party
Requesting Party Agent
Delegates-seek-authority-to
client
Client Operator
ses-
tting-to

A Data Subject may not wish to, or be capable of being, his/her own Resource Rights Administrator (for example, wishing to give power of attorney to someone else) and delegates permissions and resource management.

A Legal Person may delegate manual permissions and resource management to an administrator (for example, an employee as an Individual RRA).

Maps legal party roles to technical entity roles

Extends the "ends" to model many business relationships

Use licenses and contracts as legal devices

On the docket for (the rest of) 2018

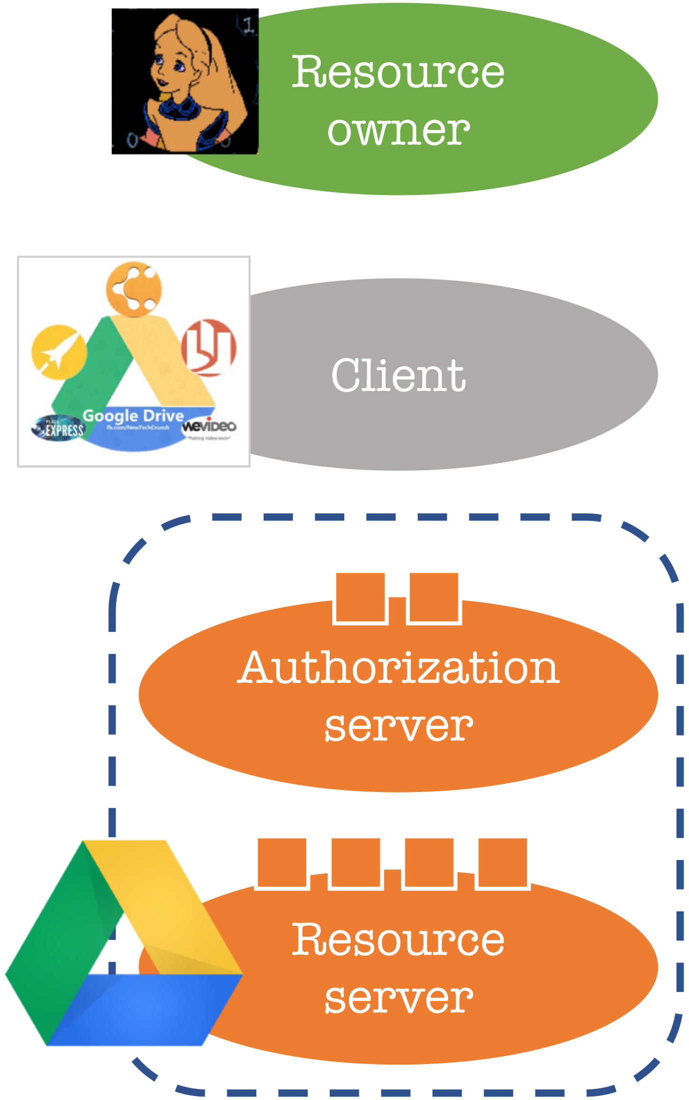
- Complete the business model and capture business scenarios
- Perform a business model POC
- Consider submitted UMA2 extensions
- Maintain UMA2 as required
- Promote UMA2 interoperability

OAuth, OIDC, and UMA2: breaking it down

Find links to UMA2 specs and swimlanes at http://tinyurl.com/umawg/

OAuth is for constrained delegation to apps

It has helped to kill the “password anti-pattern”



Resource owner

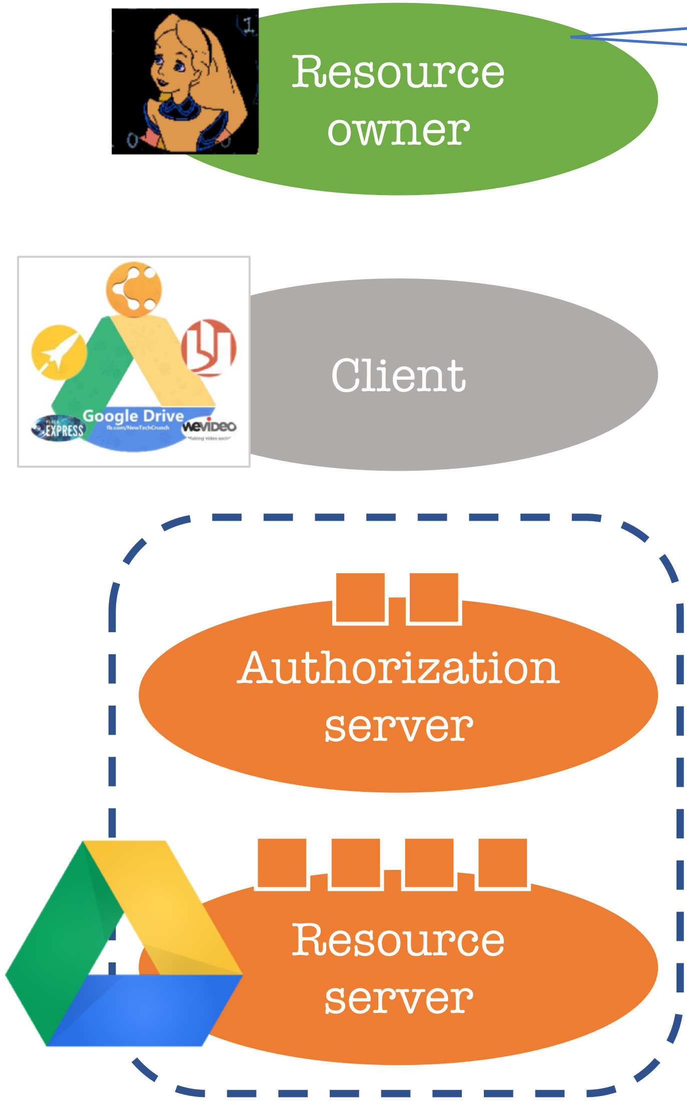
Client

Authorization server

Resource server

OAuth is for constrained delegation to apps

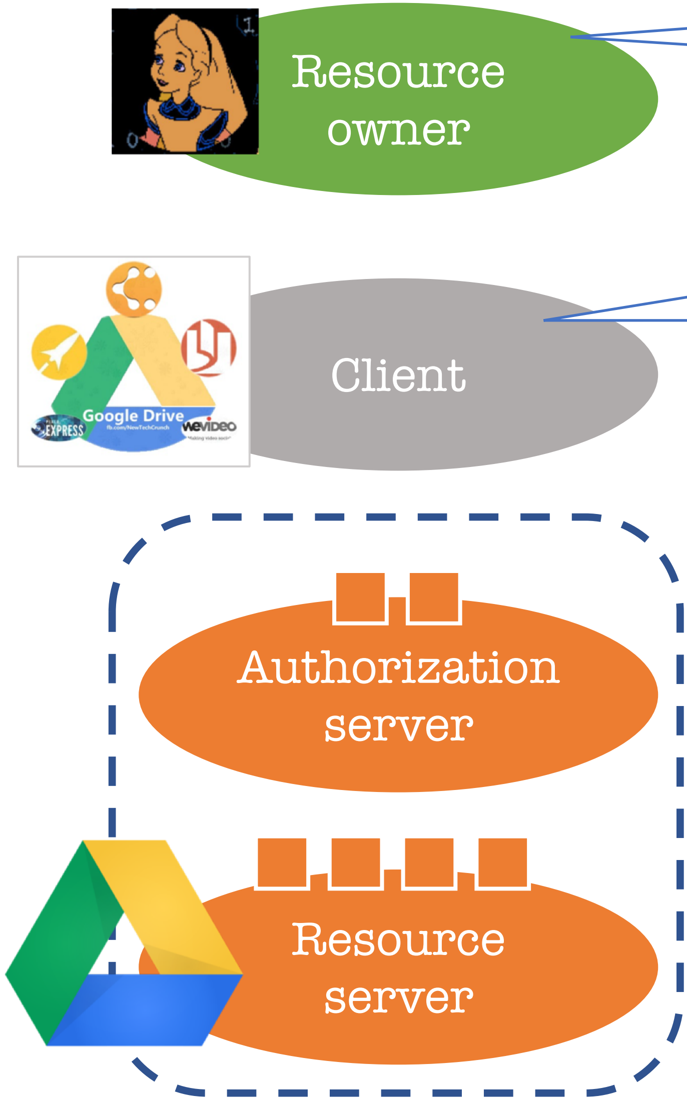
It has helped to kill the “password anti-pattern”



Authorizes (consents) at run time after authenticating, at the AS

OAuth is for constrained delegation to apps

It has helped to kill the "password anti-pattern"

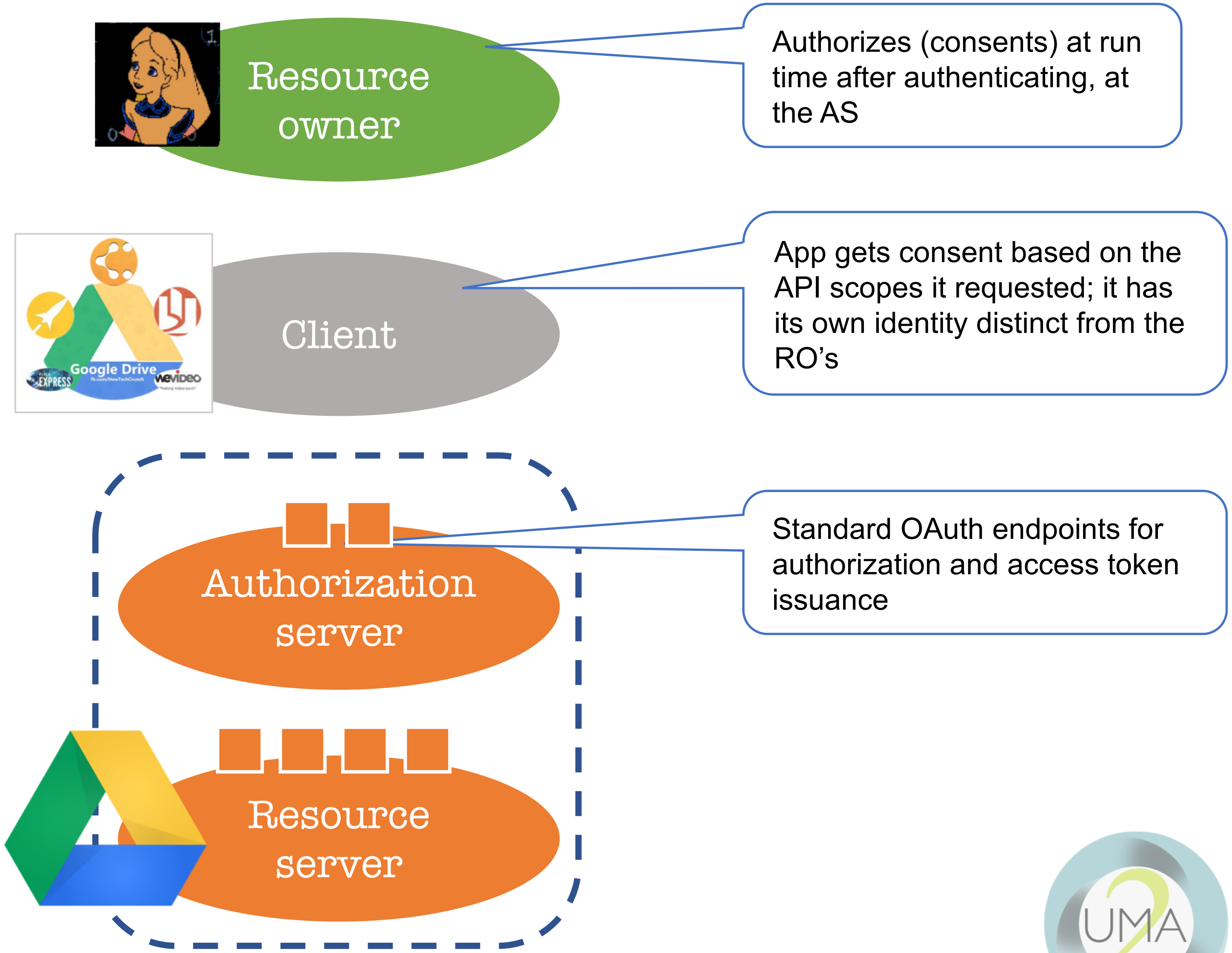


Authorizes (consents) at run time after authenticating, at the AS

App gets consent based on the API scopes it requested; it has its own identity distinct from the RO's

OAuth is for constrained delegation to apps

It has helped to kill the "password anti-pattern"



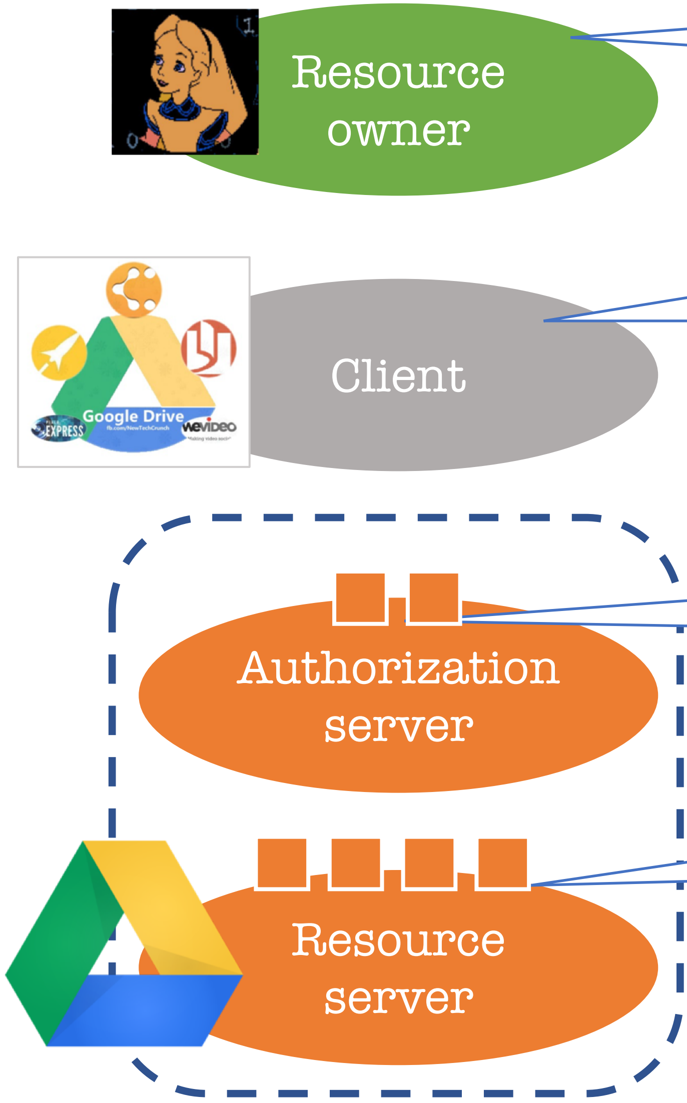
Authorizes (consents) at run time after authenticating, at the AS

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

OAuth is for constrained delegation to apps

It has helped to kill the “password anti-pattern”



Authorizes (consents) at run time after authenticating, at the AS

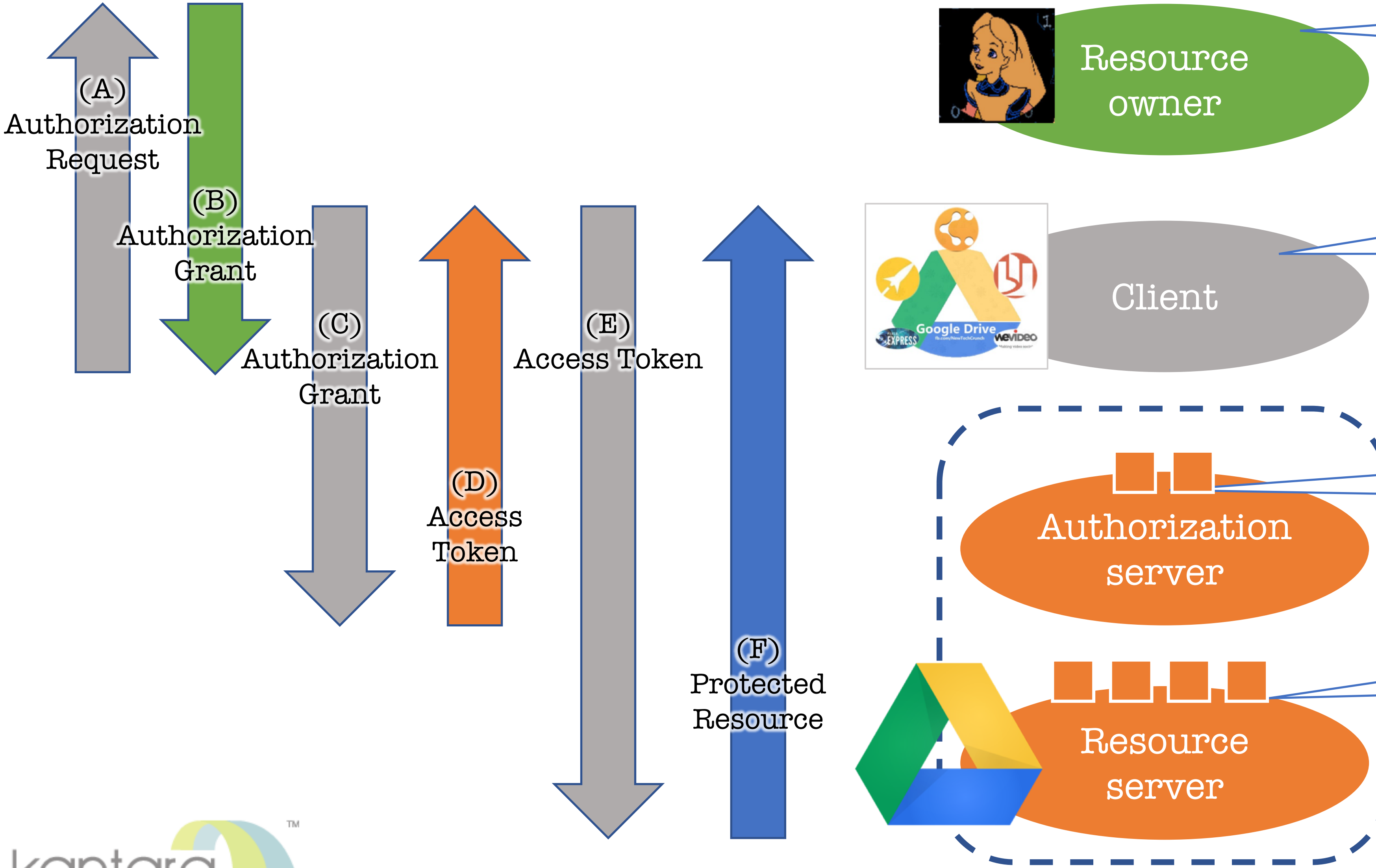
App gets consent based on the API scopes it requested; it has its own identity distinct from the RO’s

Standard OAuth endpoints for authorization and access token issuance

Some number of API endpoints that deliver the data or other value-add

OAuth is for constrained delegation to apps

It has helped to kill the "password anti-pattern"



Authorizes (consents) at run time after authenticating, at the AS

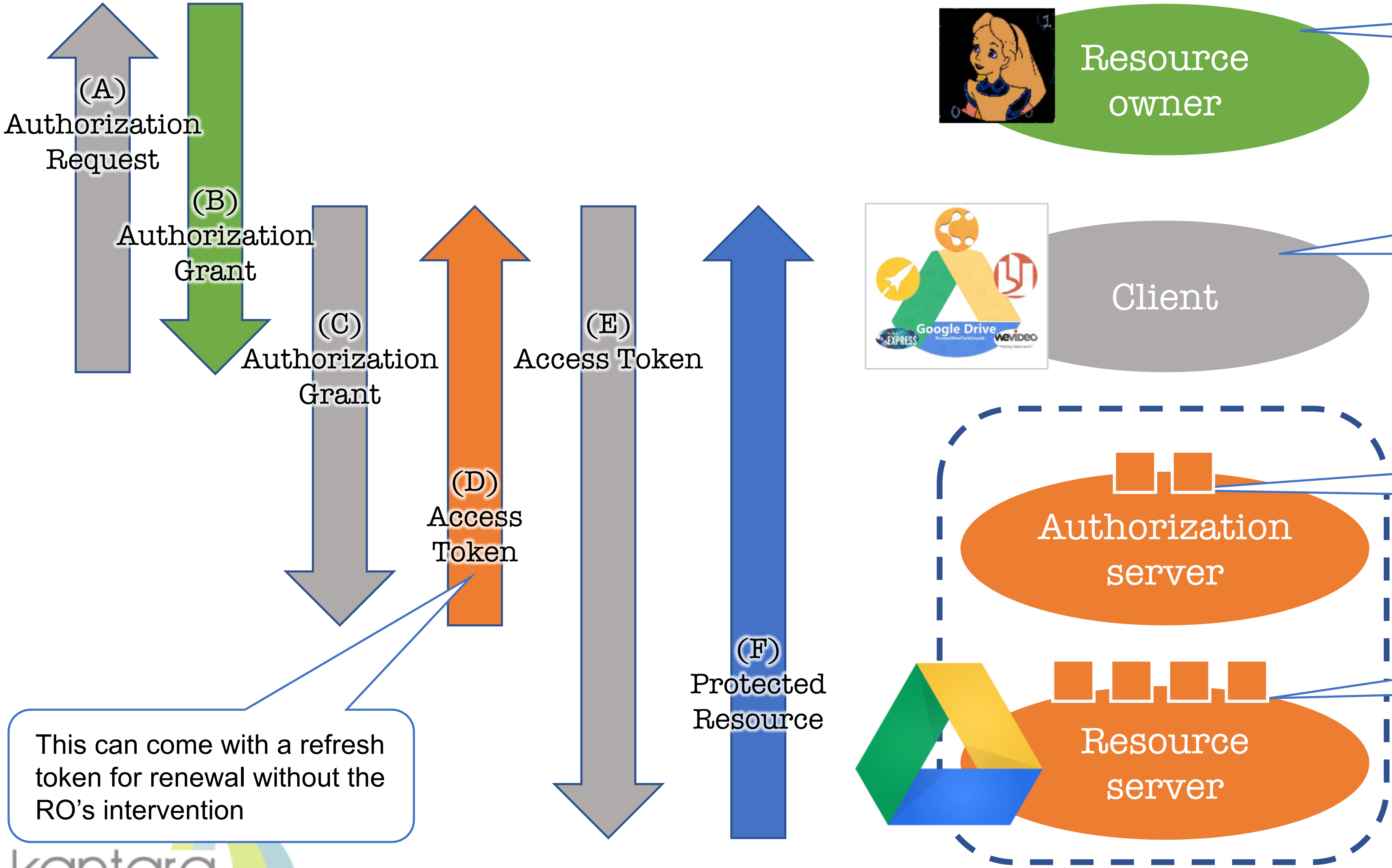
App gets consent based on the API scopes it requested; it has its own identity distinct from the RO's

Standard OAuth endpoints for authorization and access token issuance

Some number of API endpoints that deliver the data or other value-add

OAuth is for constrained delegation to apps

It has helped to kill the “password anti-pattern”



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

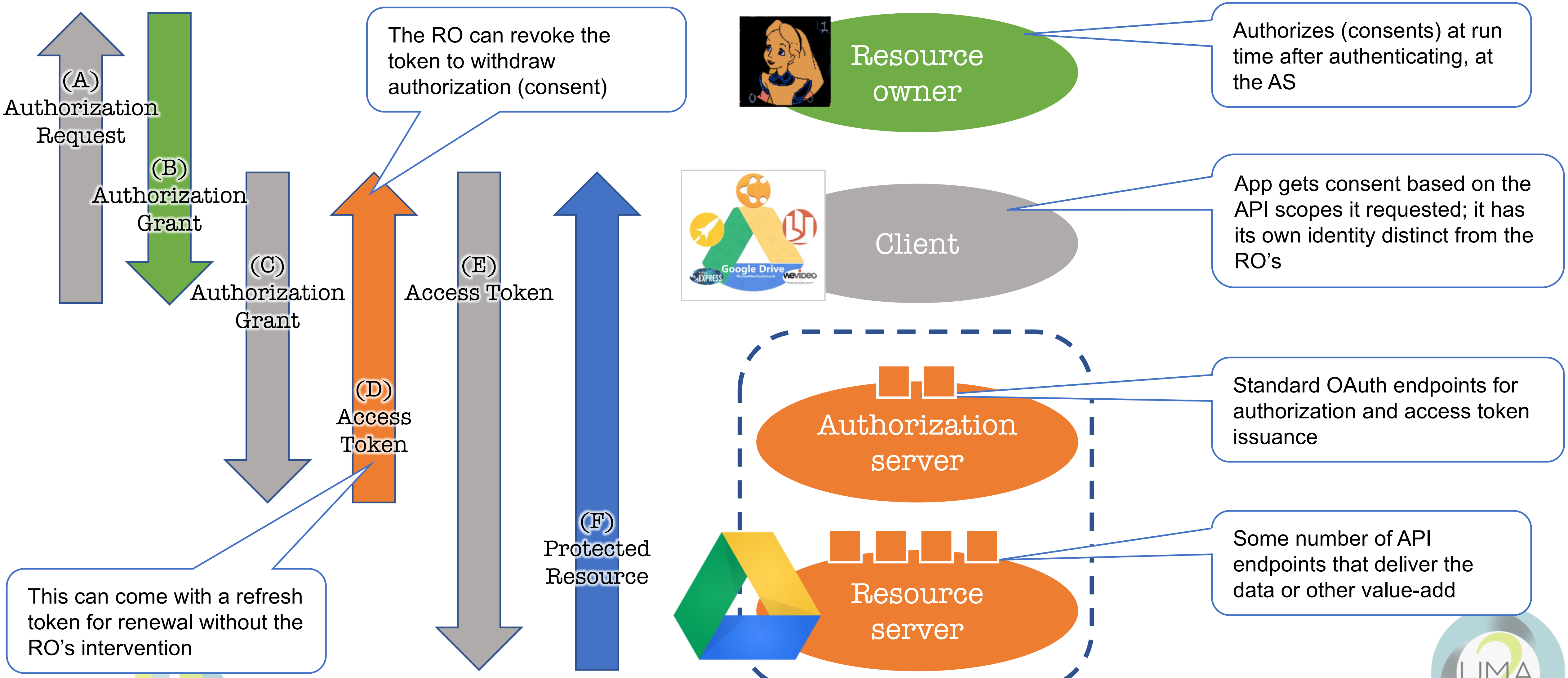
Standard OAuth endpoints for authorization and access token issuance

Some number of API endpoints that deliver the data or other value-add

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

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)

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

Authorizes (consents) at run time after authenticating, at the AS

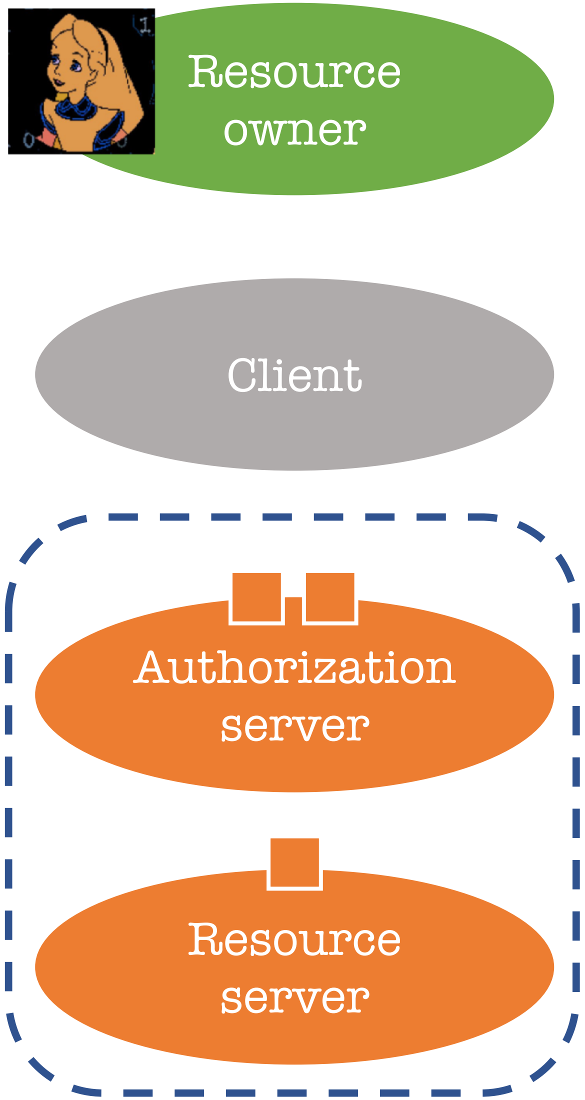
App gets consent based on the API scopes it requested; it has its own identity distinct from the RO’s

Standard OAuth endpoints for authorization and access token issuance

Some number of API endpoints that deliver the data or other value-add

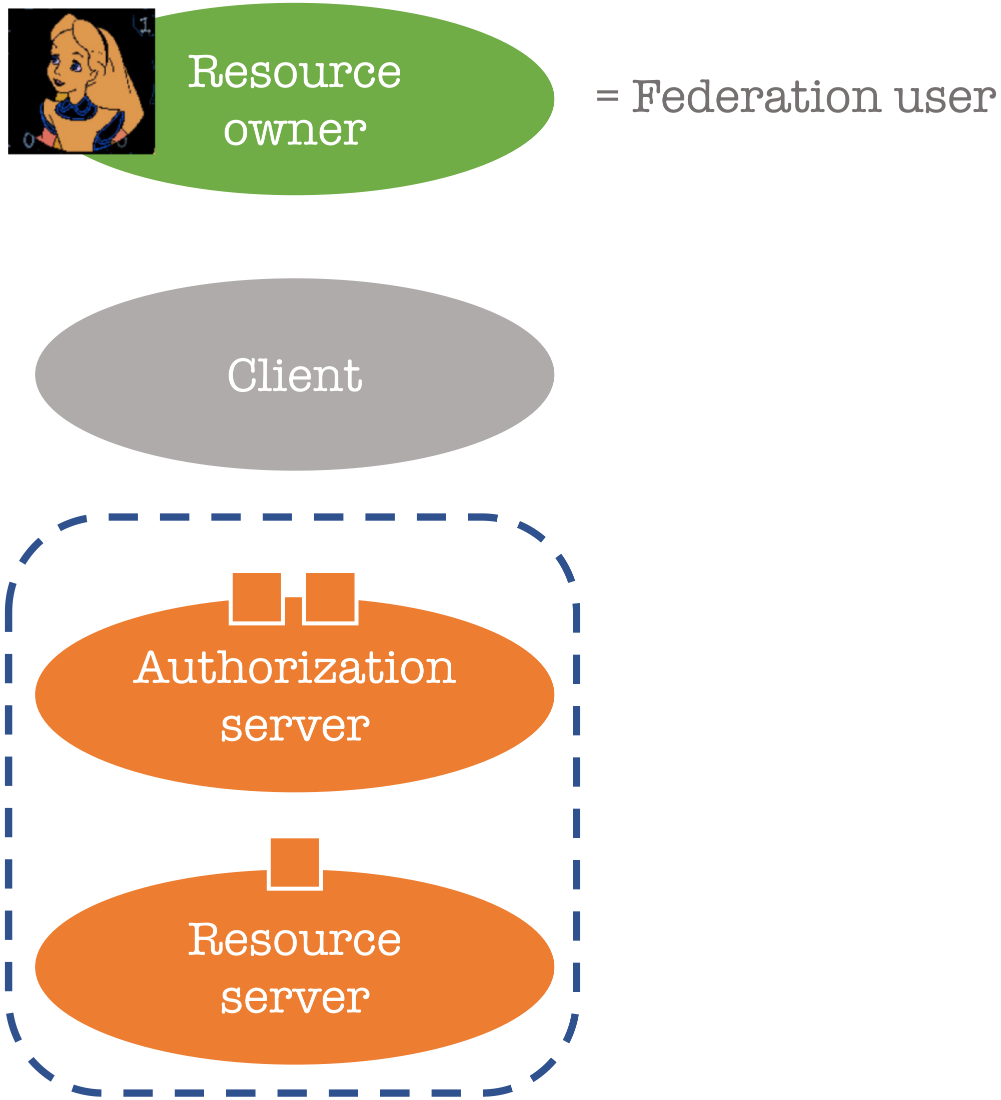
OpenID Connect does modern-day federation

It is an OAuth-protected identity API, plus a bit more



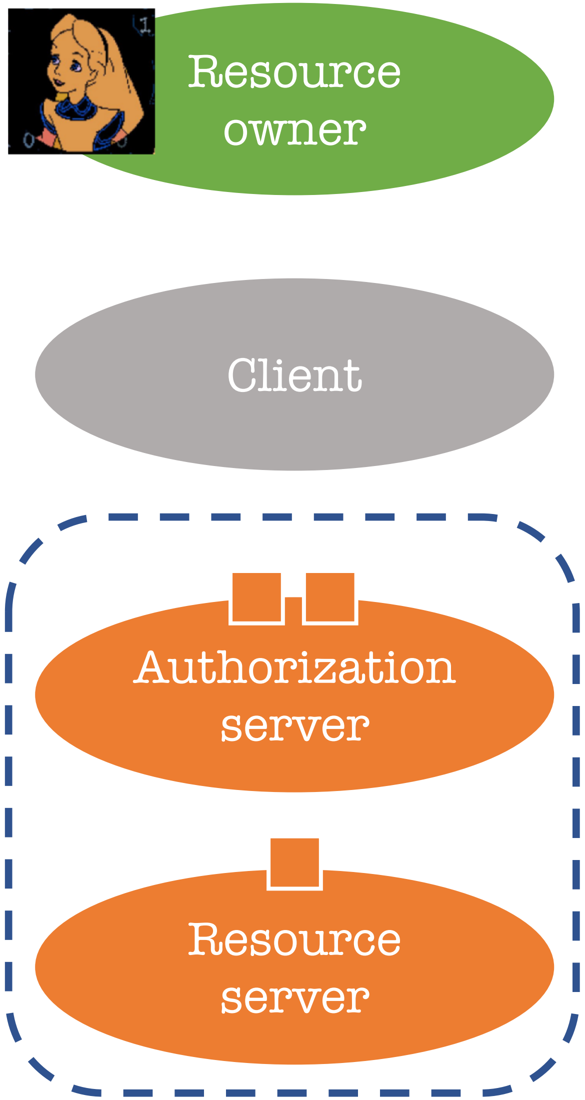
OpenID Connect does modern-day federation

It is an OAuth-protected identity API, plus a bit more



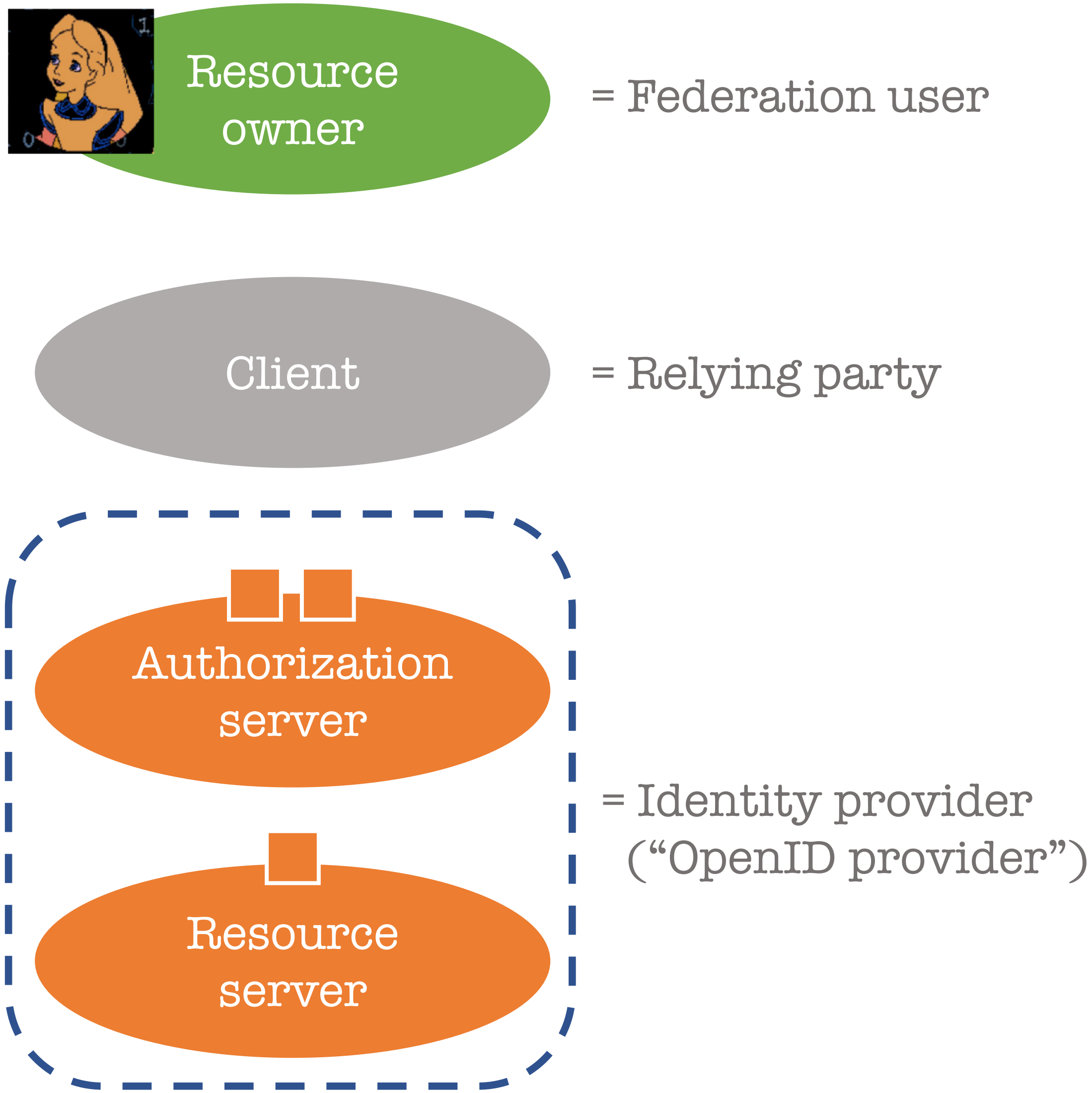
OpenID Connect does modern-day federation

It is an OAuth-protected identity API, plus a bit more



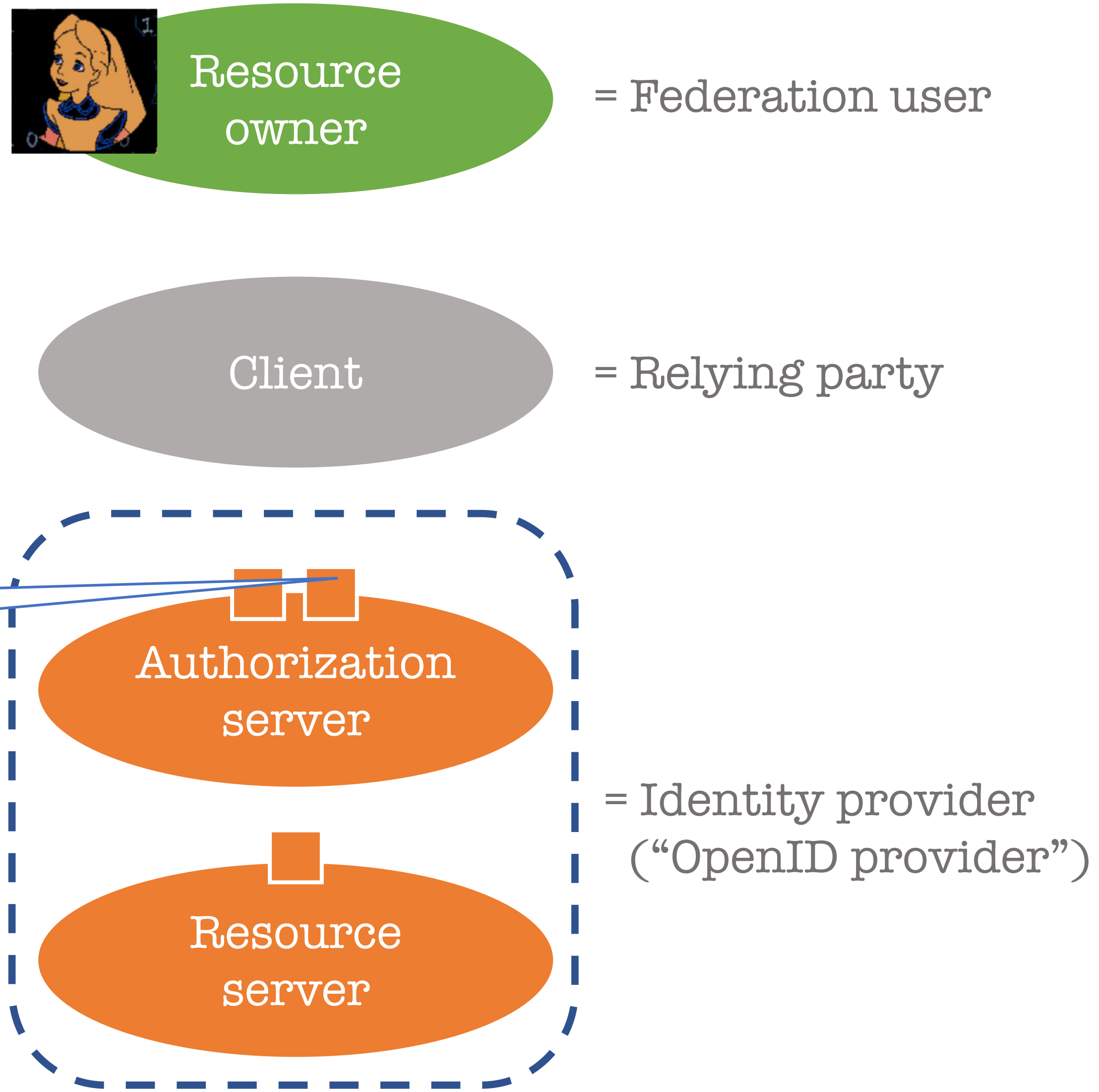
OpenID Connect does modern-day federation

It is an OAuth-protected identity API, plus a bit more



OpenID Connect does modern-day federation

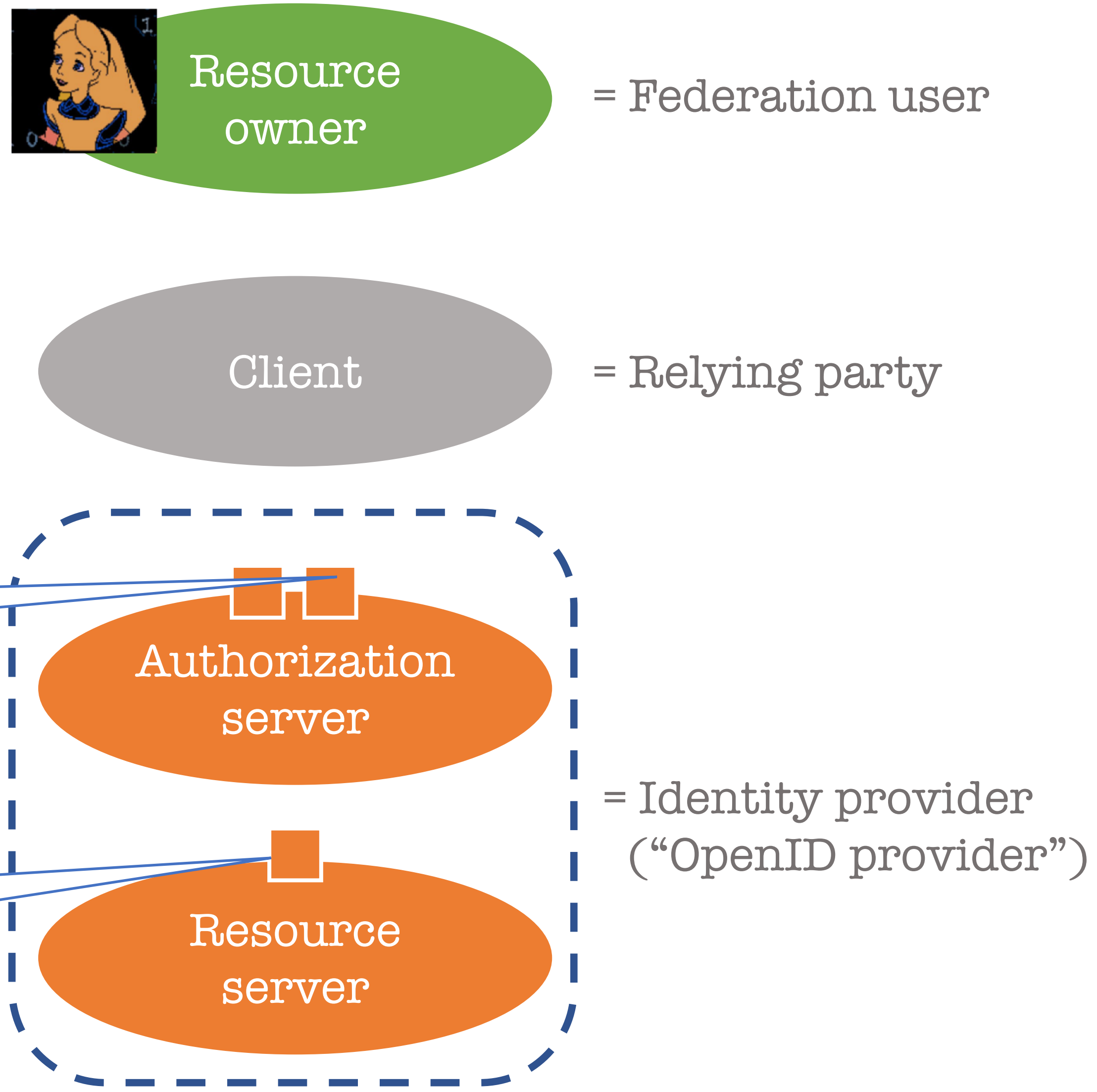
It is an OAuth-protected identity API, plus a bit more



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

OpenID Connect does modern-day federation

It is an OAuth-protected identity API, plus a bit more

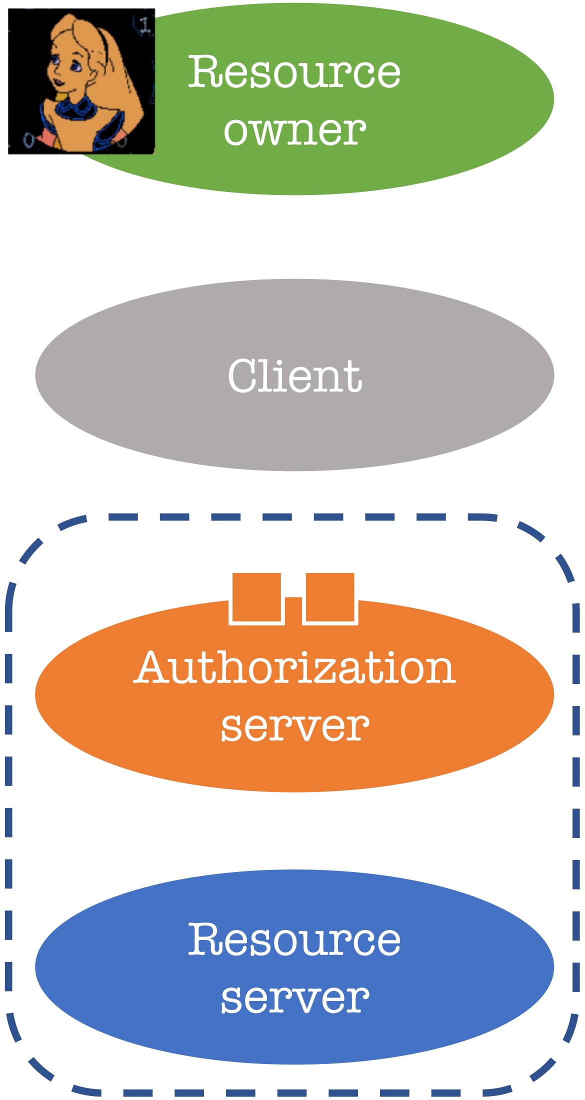


Token endpoint typically delivers an "ID token" similar to a SAML assertion

Standard UserInfo endpoint can be called with an access token to look up identity claims

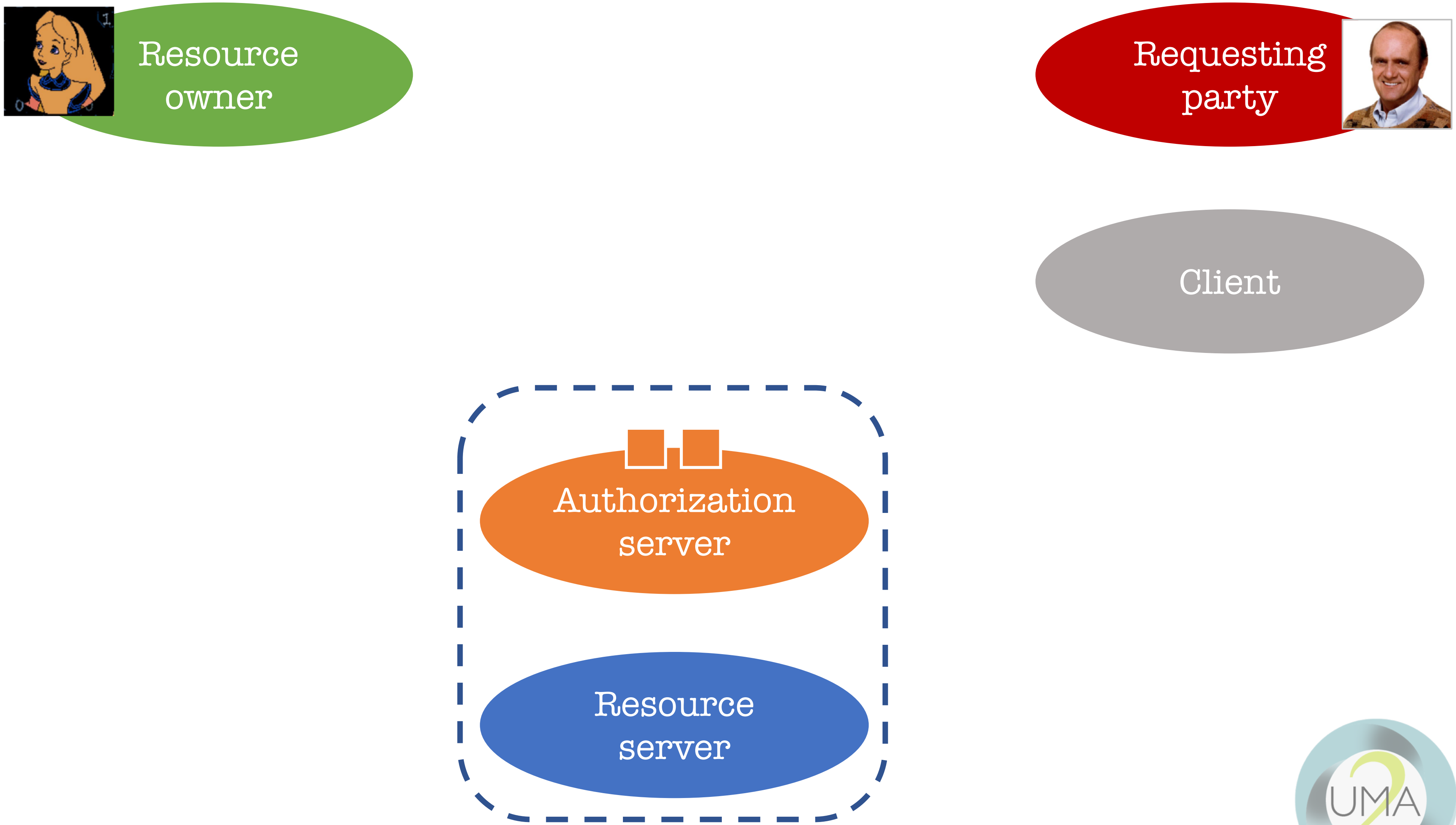
User-Managed Access is for cross-party sharing

UMA brings next-gen delegation and consent to OAuth



User-Managed Access is for cross-party sharing

UMA brings next-gen delegation and consent to OAuth



Resource owner

Requesting party

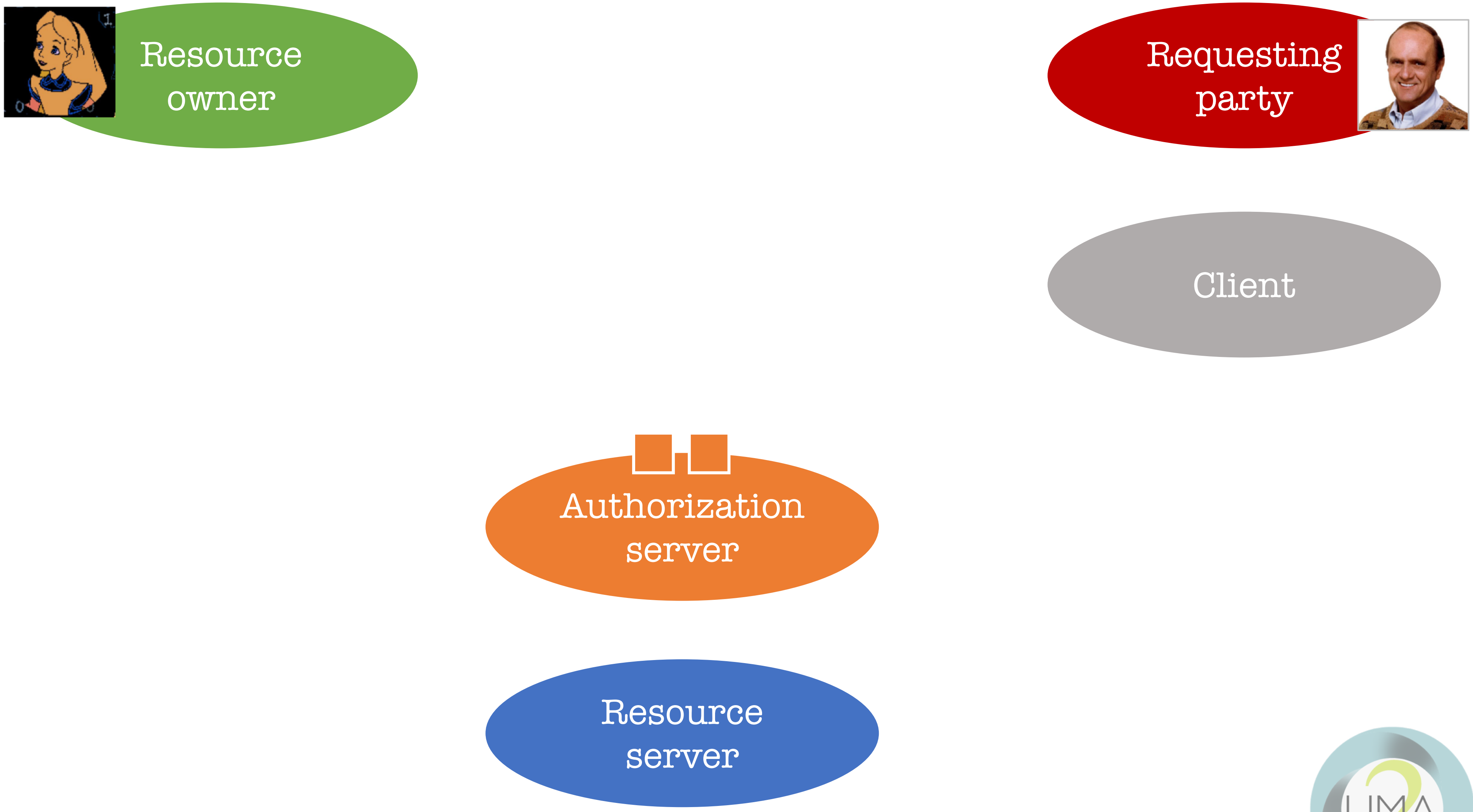
Client

Authorization server

Resource server

User-Managed Access is for cross-party sharing

UMA brings next-gen delegation and consent to OAuth



Resource owner

Requesting party

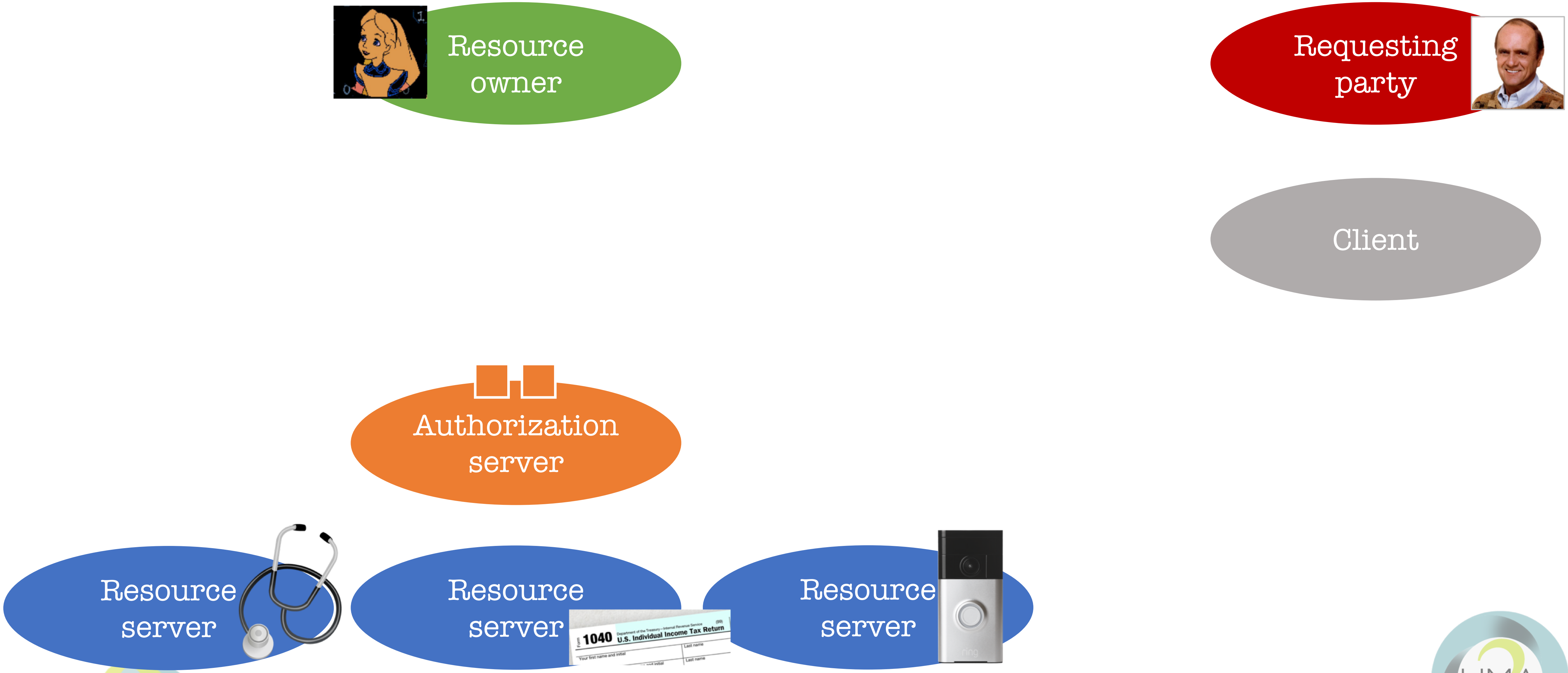
Client

Authorization server

Resource server

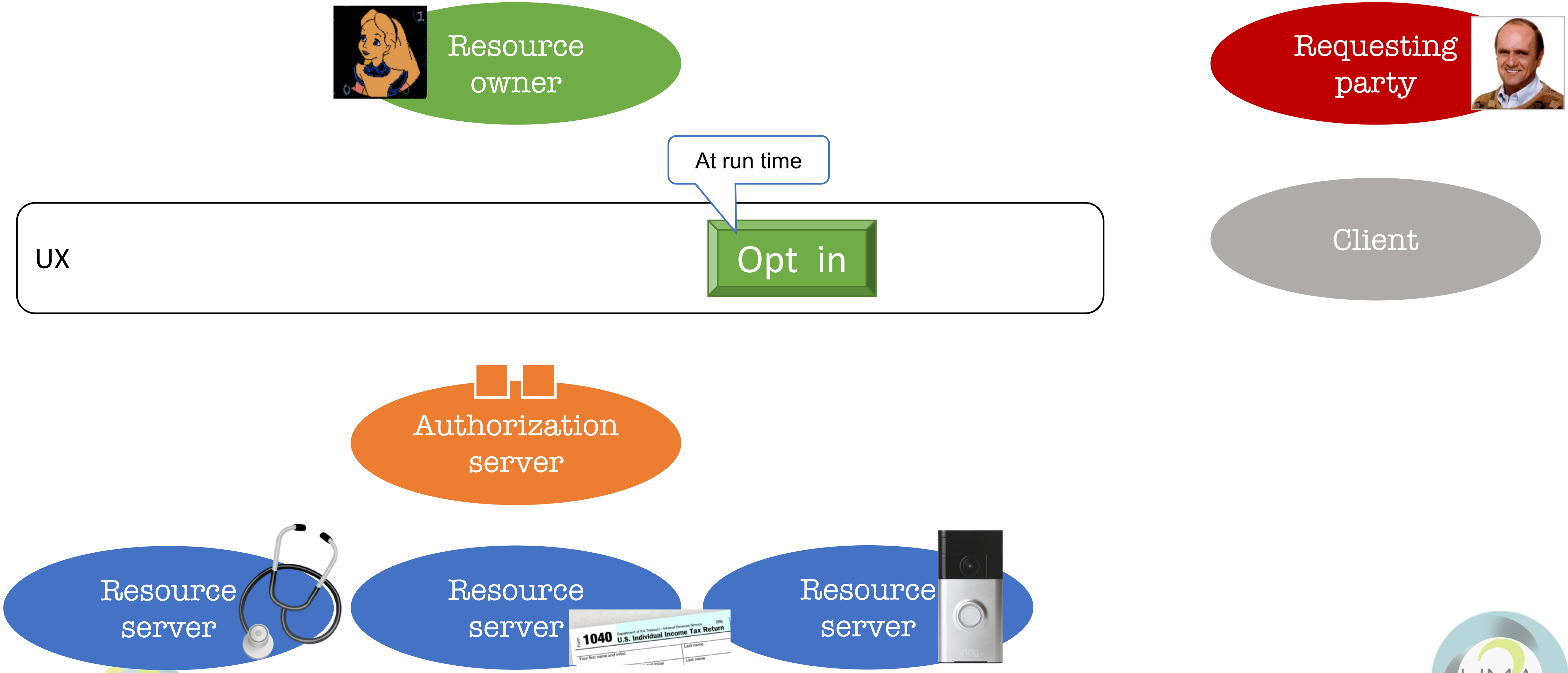
User-Managed Access is for cross-party sharing

UMA brings next-gen delegation and consent to OAuth



User-Managed Access is for cross-party sharing

UMA brings next-gen delegation and consent to OAuth



Resource owner

Requesting party

At run time

UX

Opt in

Client

Authorization server

Resource server

Resource server

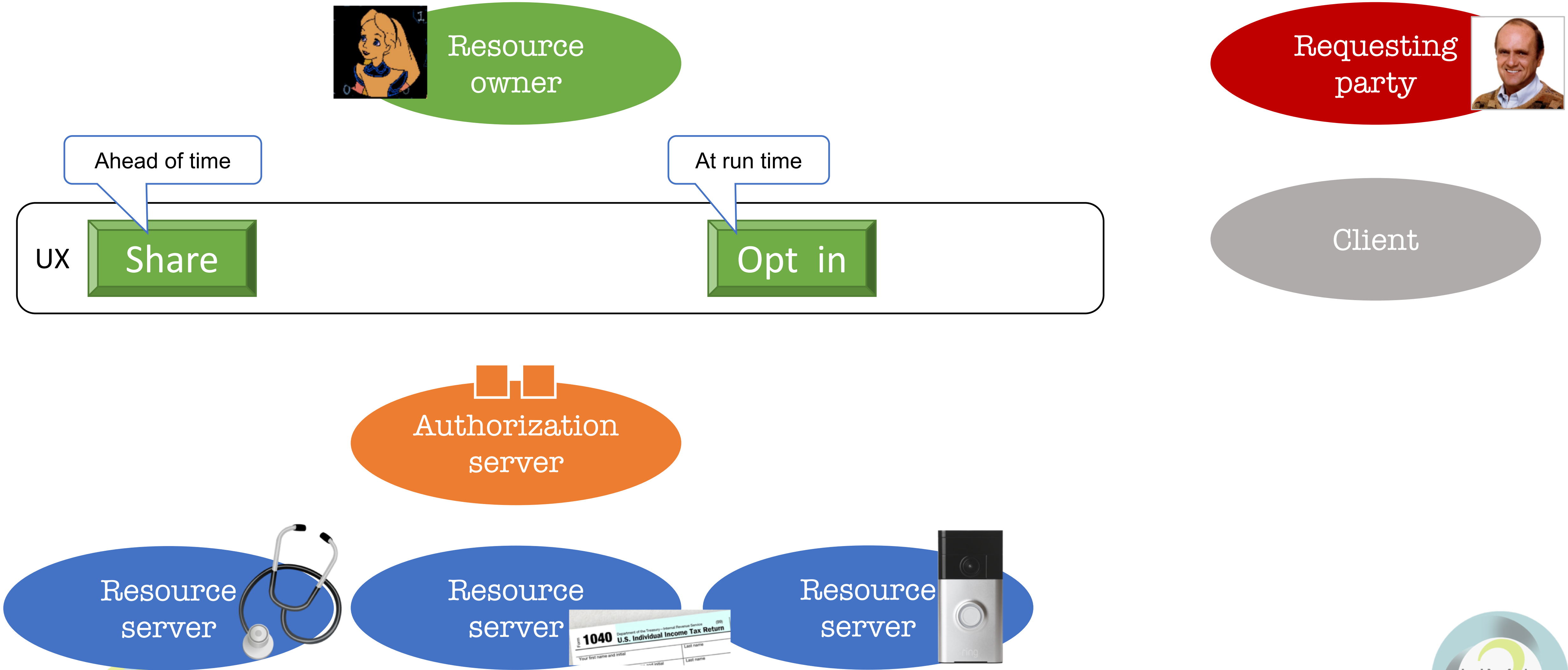
Resource server

kantara INITIATIVE

UMA 2

User-Managed Access is for cross-party sharing

UMA brings next-gen delegation and consent to OAuth

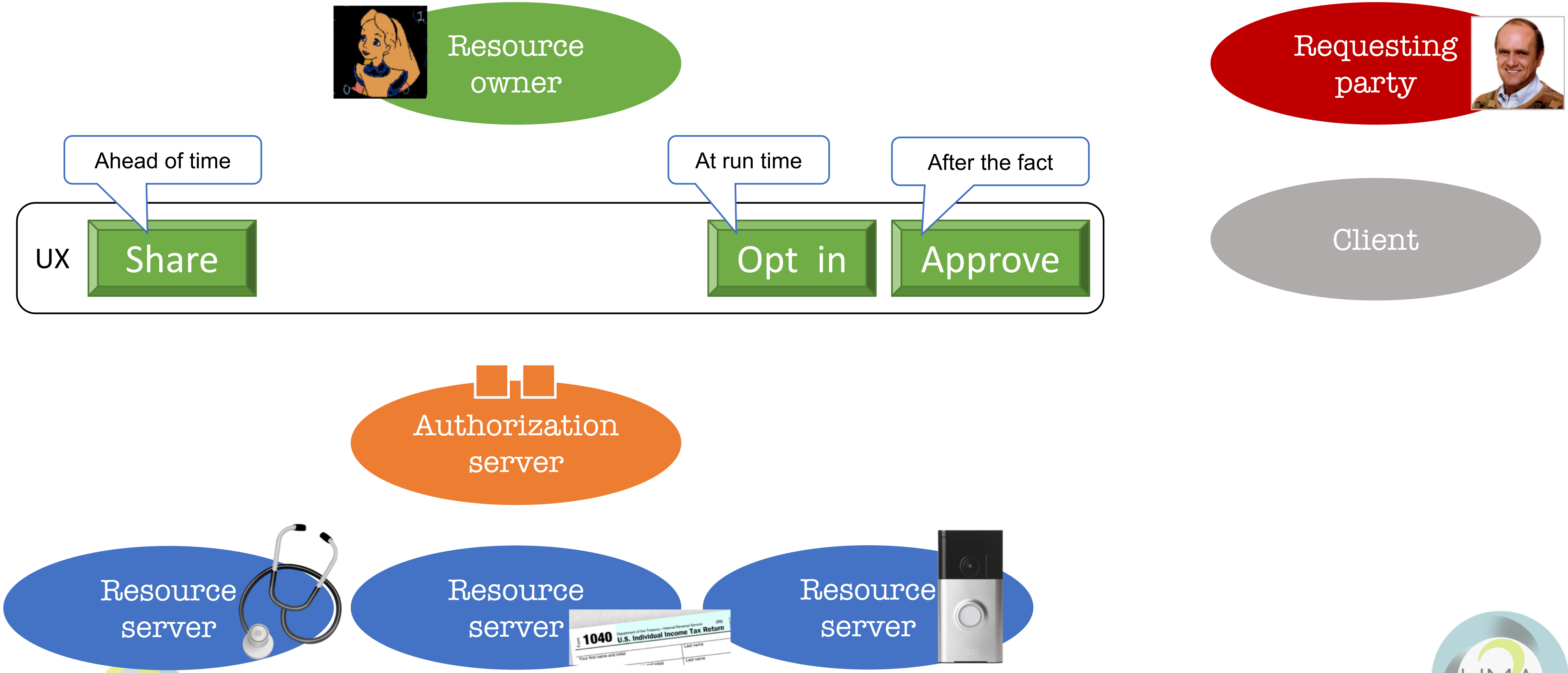


kantara INITIATIVE

UMA 2

User-Managed Access is for cross-party sharing

UMA brings next-gen delegation and consent to OAuth



Resource owner

Requesting party

Ahead of time

At run time

After the fact

UX

Share

Opt in

Approve

Client

Authorization server

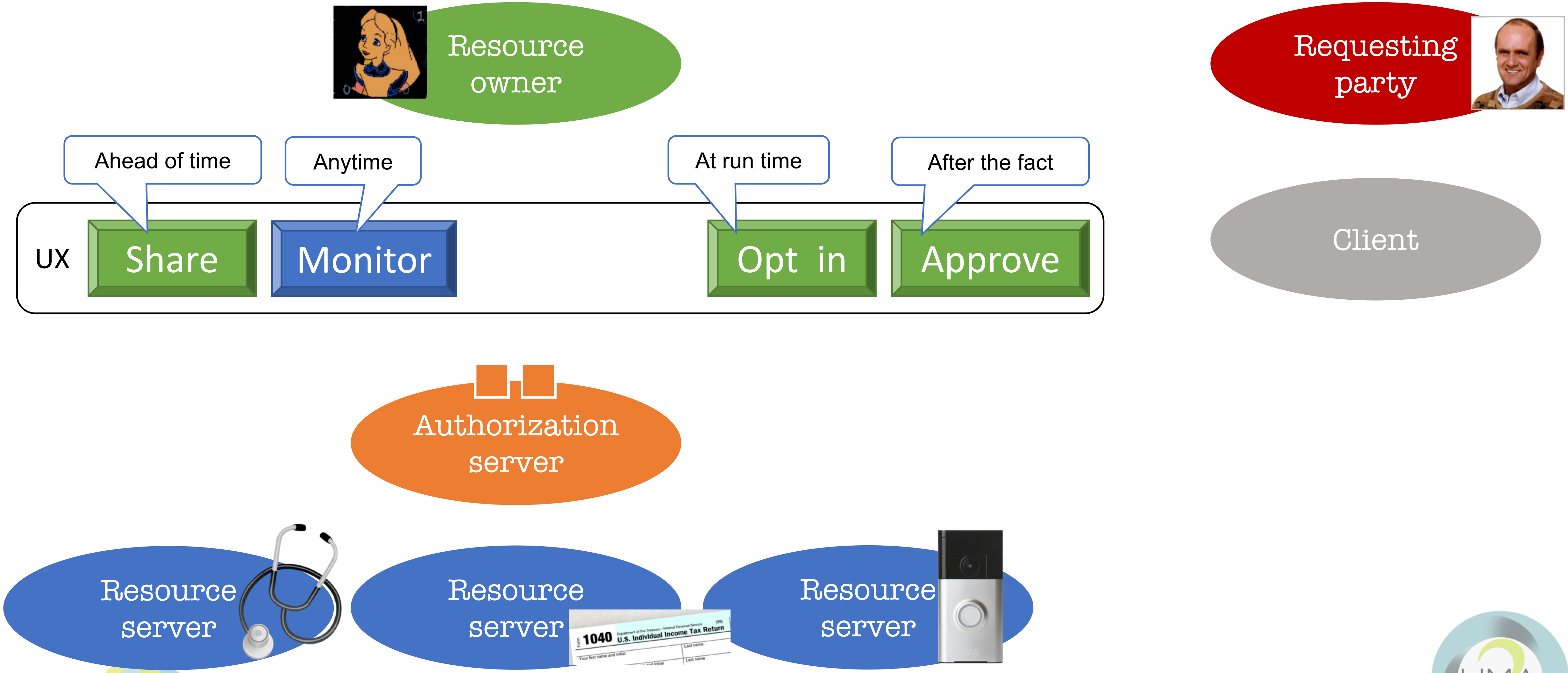
Resource server

Resource server

Resource server

User-Managed Access is for cross-party sharing

UMA brings next-gen delegation and consent to OAuth



Resource owner

Requesting party

Ahead of time

Anytime

At run time

After the fact

UX

Share

Monitor

Opt in

Approve

Client

Authorization server

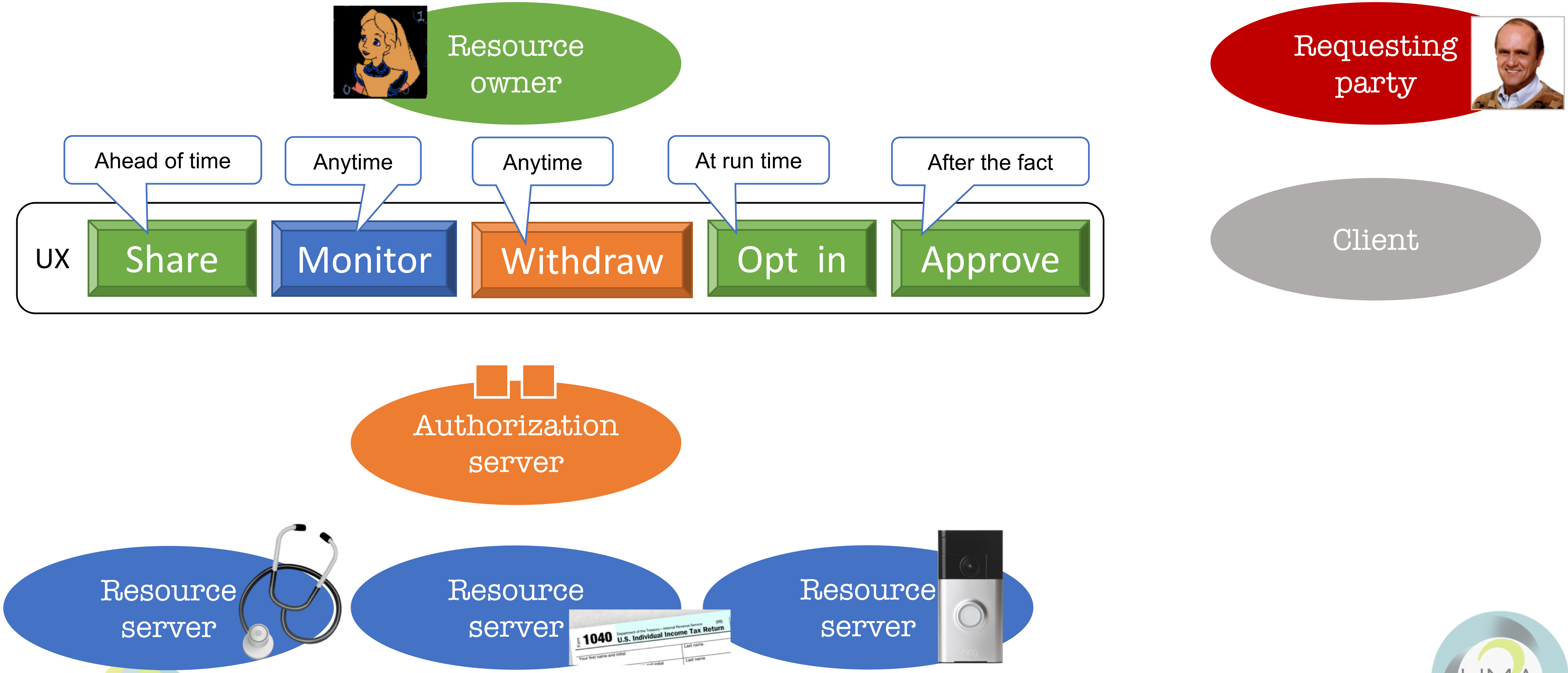
Resource server

Resource server

Resource server

User-Managed Access is for cross-party sharing

UMA brings next-gen delegation and consent to OAuth



Resource owner

Requesting party

Ahead of time

Anytime

Anytime

At run time

After the fact

UX

Share

Monitor

Withdraw

Opt in

Approve

Client

Authorization server

Resource server

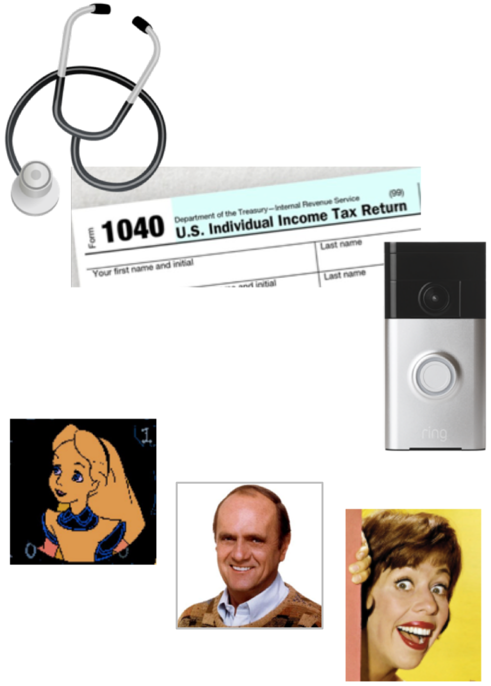
Resource server

Resource server

Key benefits of UMA to service providers

True security of delegated access

Scalability of resource permissioning

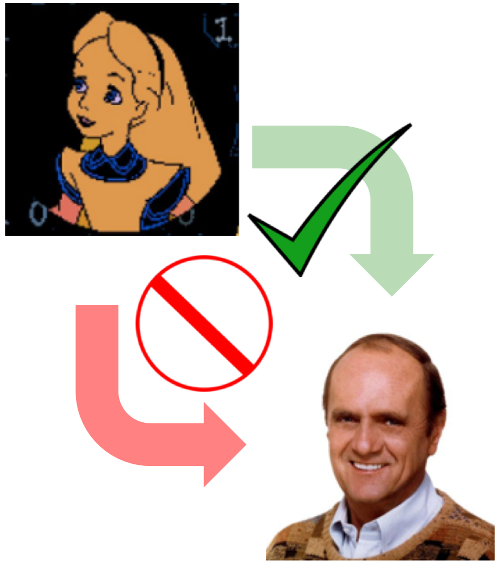

1040
Department of the Treasury—Internal Revenue Service
U.S. Individual Income Tax Return
Your first name and initial
Last name
ring

API-first protection strategy

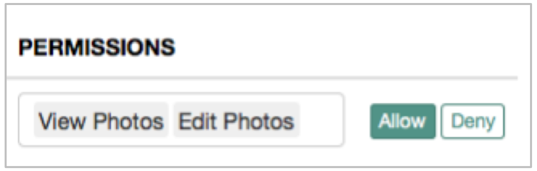
Fosters control for compliance and trust

Key benefits of UMA to consumers

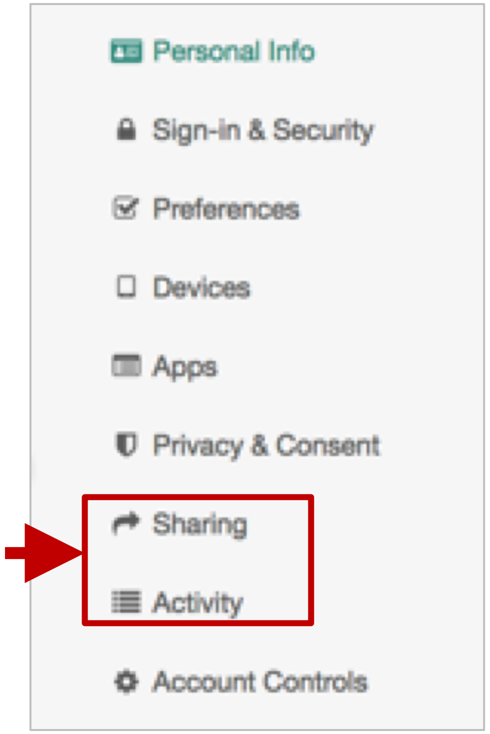
Constrained party-to-party delegation



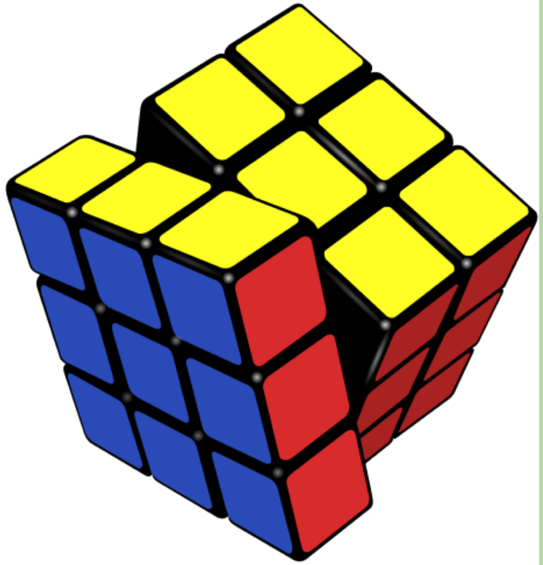
Granting consent without external influence



Centralized monitoring and management



Control of consents at a fine grain



Questions?
Thank you!
Join us!

Eve Maler, WG chair

eve.maler@forgerock.com| @xmlgrrl

15 May 2018

http://tinyurl.com/umawg/ | @UMAWG