UMA2 Legal role definitions

Some visualizations

(See the companion draft report *A Proposed Licensing Model for User-Managed Access*, available at: [https://kantarainitiative.org/reports-recommendations/](https://kantarainitiative.org/reports-recommendations/))
Formal UMA business model
Legal relationships: Basic conventions

Color conventions:
- **Legal** (upper capital, blue)  technical (lowercase, orange) */issue/question/note (red)

Basic relationship types:
- Party role A is a kind/species of party role B: **Is-a** (for more detail, see Business Model definitions appendix)
- Party role A acts in technical entity role A: **Acts-as-a** (maps party defined in Business Model to technical entity defined in specs)
Legal relationships: Persons

Establishing basic party roles: Individual, Legal Person, Data Subject

Individual: A natural Person.
Reference: Users-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization and supporting documentation; UCITA action 102(a)(51); UETA action 2(12); RUF/ADA 2(17).

Legal Person: A corporation, business trust, estate, trust, partnership, limited liability company, association, joint venture, governmental subdivision, instrumentality, or agency, public corporation, or any other legal or commercial entity.
Reference: Users-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization and supporting documentation; UCITA action 102(a)(51); UETA action 2(12); RUF/ADA action 2(17).

Data Subject: The Person to whom a Protected Resource relates.
Reference: Users-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization and supporting documentation; RUF/ADA action 2(21) (“Protected Person”).

Client Operator: A Person responsible for running and operating a software application (the “Client”) used by a Requesting Party or Requesting Party Agent to access and use a Protected Resource.
Reference: Users-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization and supporting documentation.

Requesting Party: A Person with legal capacity and authority, either as an Individual or Legal Person, to request and secure access to a Protected Resource either directly with a Resource Server Operator or by means of a Client Operator.

Requesting Party Agent: A Person seeking access to a Protected Resource on behalf of a Requesting Party and by means of a Client software application.
Reference: Users-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization and supporting documentation.
Legal relationships: Legal-to-technical role bridges
Establishes how parties in legal roles can take part in UMA messaging flows

- **Resource Rights Administrator** acts as a resource owner.
- **Authorization Server Operator** acts as an authorization server.
- **Resource Server Operator** acts as a resource server.
- **Client Operator** acts as a client.
- **Requesting Agent** acts as a requesting party.

---

**resource owner**
An entity capable of granting access to a protected resource, the “user” in User-Managed Access. The resource owner MAY be an end-user (natural person) or MAY be a non-human entity treated as a person for limited legal purposes (legal person), such as a corporation.

**requesting party**
A natural or legal person that uses a client to seek access to a protected resource. The requesting party may or may not be the same party as the resource owner.
Legal relationships: Business relationship types

- Delegates authority for granting and managing access permissions to: Delegates-perm-authority-to
  - aka Agency Contract
- Delegates resource management to: Delegates-mgmt-to
  - aka Access Contract
- Licenses granting access permissions to: Licenses-perm-granting-to
- Licenses receiving access permissions to: Licenses-perm-getting-to
- Delegates access seeking authority to: Delegates-seek-authority-to
- Delegates permission to know/persist to: Permits-knowing-claims
- Party in role A also acts in role B: Acts-as-a
Legal relationships: “Endpoint to endpoint”

*Intra-protocol relationships among parties in legal roles, illustrated*

- **Resource Rights Administrator** (Agency Contract)
  - Delegates-perm-authority-to
  - Delegates-mgmt
- **Authorization Server Operator**
  - Licenses-perm-granting-to
- **Resource Server Operator** (Access Contract)
  - Permits-knowing-claims
- **Client Operator**
  - Delegates-seek-authority-to
- **Requesting Agent**
  - Licenses-perm-getting-to
  - Delegates-seek-authority-to
Legal relationships: “Endpoint to endpoint” - Flattened

Intra-protocol relationships among parties in legal roles, illustrated

- Resource Rights Administrator
- Requesting Agent
- Resource Server Operator
- Authorization Server Operator
- Client Operator

Relationships:
- Delegates-Perm-authority-to
- Delegates-mgmt-to
- Licenses-Perm-getting-to
- Licenses-Perm-granting-to
- Permits-knowing-claims
- Licenses-Perm-getting-to
- Delegates-seek-authority-to
- (Agency Contract)
- (Access Contract)
Legal relationships: “Extending the ends”
How “offline” types of parties may play delegation roles, illustrated

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).

NOTE: It is not currently in the scope of the UMA Legal subteam to consider such use cases.

A Requesting Party may be an Individual or a Legal Person, and a Requesting Agent may also be an Individual or a Legal Person. The former may not wish to be, or be capable of being, its own Agent.

In that case, a Requesting Party may delegate access-seeking authority to another party on its behalf (for example, in the case of a hospital having a specific clinician seek access as its employee, an Individual RqPA).

(Agency Contract)
(Access Contract)
Legal relationships: Devices and artifacts

Making relationships and their changes auditable and machine-readable

Resource server
Authorization server

Resource Rights Administrator
(Agency Contract)

Delegates-perm-authority-to
(Access Contract)

Delegates-mgmt-to

Authorization Server Operator

Resource Server Operator

Authorization Server Operator

Resource Server Operator

Legal devices: ToS, privacy notice (when an Individual)

Technical artifacts: Consent management record (outside UMA scope)

Licenses-perm-granting-to

Legal devices: OAuth client agreement (prior to resource owner context -- licensing needs to be set up there)

Technical artifacts (type): Client credentials for UMA resource server (OAuth); PAT (UMA); all request/response messages between authorization server and resource server (UMA)

1

2

3
Legal relationships: Devices and artifacts

Making relationships and their changes auditable and machine-readable

Authorization Server Operator

Client Operator

Requesting Agent

Licenses- perm-getting-to

License builds on previous devices and is carried through technical artifacts

Technical artifacts (type):
- Claim token (UMA)
- Authorization server/requesting party request/response messages (UMA)

licenses- perm-getting-to

Technical artifacts (type):
- Client credentials for UMA client (OAuth)
- RPT (UMA)
- all authorization server/client request/response messages (UMA)
- Policies (outside the scope of UMA)

Legal devices:
OAuth client agreement for UMA client (prior to requesting party context -- licensing needs to be set up there)

Legal devices:
License builds on previous devices and is carried through technical artifacts

Technical artifacts (type):
- PCT (UMA)
- all authorization server/requesting party request/response messages (UMA)

Legal devices:
OAuth client agreement for UMA client

Technical artifacts (type):
- Client credentials for UMA client
- RPT (UMA)
- claim token (UMA)
- all authorization server/client request/response messages (UMA)
- Policies (outside the scope of UMA)
Legal relationships: Devices and artifacts

Making relationships and their changes auditable and machine-readable

Authorization Server Operator

Client Operator

Requesting Agent

Permits-knowing-claims

Delegates-seek-authority-to

authorization server

Legal devices: Possibly ToS/privacy notice (when an individual); note that this is not the requesting side’s but the resource-owning side’s authorization server

Technical artifacts (type): PCT (UMA), all requesting party/authorization server request/response messages (These are all front-channel messages; what are options for auditing?)

Legal devices: ToS, privacy notice (when an Individual)

Technical artifacts (type): Claim token (UMA), PCT (UMA), all requesting party/client/authorization server request/response messages (Authorization server issues the PCT, which may capture requesting party’s authorization/consent; client holds PCT. How to reflect this?)
Legal relationships: Devices and artifacts

Making relationships and their changes auditable and machine-readable

Legal devices: Law or contract
Technical artifacts: Outside UMA scope
Legal relationships: One-party/multi-role scenario patterns

*In some cases...*

- **Data Subject** acts as a **Resource Rights Admin**
  - Alice is controlling access to her own protected resources, vs. newborn/incompetent/etc. Johnny’s.
  - Alice has built/is running her own “personal authorization server”. See HIE of One.

- **Resource Rights Admin** acts as a **Authorization Server Operator**
  - A variant where Alice is running a PAS for Johnny.
  - Alice has built/is running a “personal data store” for herself.

- **Data Subject** acts as a **Resource Server Operator**
  - A variant where Alice is running a PDS for Johnny.

- **Resource Rights Admin** acts as a **Resource Server Operator**

- **Requesting Party** acts as a **Requesting Agent**
  - Bob is seeking access on behalf of himself, instead of doing it as “work for hire” on behalf of an employer.
  - The ultimate party seeking access has built/is running their/its own client application (could be an individual or legal person).

- **Requesting Party** acts as a **Client Operator**
  - A variant where this is true of the agent of the ultimate party seeking access. (Included here for completeness but may be too detailed?)

- **Requesting Agent** acts as a **Client Operator**

- **Data Subject** acts as a **Requesting Party**
  - The same Person seeking access is the one whose resources are being protected. This is a typical OAuth scenario. (There are more “Agent” variants.)
Legal relationships: More scenario patterns

In some cases...

- Acts as a **Authorization Server Operator**
  - Acts-as-a
  - Resource Server Operator
    - resource server
  - Acts-as-a
  - Authorization Server Operator
    - Authorization Server Operator
    - Acts-as-a
    - Client Operator
      - client
    - Acts-as-a
    - Identity Provider
  - Acts-as-a
  - Identity Provider
  - Acts-as-a
  - Resource Server Operator
  - Acts-as-a
  - Client Operator
    - client
    - Acts-as-a
    - Identity Provider
  - Acts-as-a

...and ASO **runs all available resource servers**. This relatively tighter ecosystem is consistent with how most OAuth deployments are run; it may still be interested in exposing the UMA Federated Authorization (protection API) interface for auditability reasons.

*...and ASO **runs all available clients**. This tighter ecosystem (possibly in combination with the above) may still be interested in having the authorization server expose the various UMA interfaces for auditability reasons.*

There are a variety of deployment options possible for sourcing resource owner identity (and requesting party claims). A business layer such as a trust framework can take into account identity assurance, authentication, and claims requirements. (*"Identity Provider" is not an UMA-related party role and UMA is agnostic as to identity, identification, and authentication.*)
Scenario: Cradle-to-grave

1. Data Subject is too young to use digital assets

Data Subject is newborn Johnny. Resource Rights Administrator is mother Alice. Delegation from DS to RRA is by law in this case because she is his legal guardian. She manages his protected resources (personal data/digital assets) online and grants access to others on his behalf, for the period that she is his guardian. Alice may selectively grant access to Johnny’s protected resources, such as EHR data and school records, to caregivers, family members, nannies, and others. These parties may be acting as individuals or on behalf of larger organizations/institutions, and be using a variety of client applications.

*Some relationship lines have been removed for clarity.*
Scenario: Cradle-to-grave

2. Data Subject is old enough to use assets but too young to consent to their use

Data Subject Johnny grows old enough to begin using online services. Resource Rights Administrator Alice begins to give some control of his resources (personal data/digital assets) to him. One way to handle this is by enabling Alice to grant access to Johnny’s own resources to him as a Requesting Party Agent on his own behalf as a Requesting Party. (In certain jurisdictions, a verified citizen identity may have been created for him at birth or at a young age, which he could claim and use now.)
Scenario: Cradle-to-grave
3. Data Subject is old enough to consent to their use and manages digital assets themselves

Data Subject Johnny is old enough to need a legal guardian no longer and may even wish to withdraw his own mother (former Resource Rights Administrator) Alice’s access to his resources (personal data). This may be true at least for certain resources, possibly based on standardized data types, correlated to jurisdictional law. For a start, the relevant delegations to her could be rescinded, which cascades into revoking relevant UMA tokens, likely policies, and other artifacts and replacing Alice as the resource owner with himself. (Such UMA “molecular bond” rearrangements are not part of UMA per se, but could be part of an identity relationship management automation layer.)
Scenario: Cradle-to-grave

3a. Steady state: Data Subject manages their own digital assets

In the typical case, Data Subject Johnny will manage his digital assets as his own Resource Rights Administrator. He can share access with others as he sees fit.
Scenario: Cradle-to-grave

4. There are multiple administrators of resource rights

There may be several Resource Rights Administrators, either because they map to multiple Data Subjects (as in the case of joint bank accounts or genomic data) or because a single Data Subject has delegated resource rights administration to multiple other or additional parties (for example, holders of power of attorney). Identity Relationship Management is required to ensure that distinct resource IDs for each resource owner are treated as "the same virtual resource" in some fashion, and that RRA entrances and exits are tracked.
Scenario: Cradle-to-grave

5. Data Subject becomes mentally incapacitated or dies

Going by the RUFADAA pattern:

a) The Data Subject doesn’t designate anyone ahead of time to manage their digital assets; the assets’ Custodian (Resource Server Operator) then becomes the Designated Recipient of access (RRA) per its terms of service.

Or b) the DS creates a disclosure permission plan to designate their Personal Representative to manage their digital assets. That representative can ultimately share with whomever else as a Requesting Agent.

Or c) the DS makes someone other than their Personal Representative a Designated Recipient of resource rights administration. The RRA in turn may, or may not, grant access to the Personal Representative.
The Pensions Dashboard project is a government fintech initiative for the UK consumer. The Origo solution is securely identifying the consumer before orchestrating a search of pensions created in previously held jobs across the industry. “Wee Alice” (acting as her own DSA) first grants pension access to an LOA version of herself, “Big Alice”. The government runs the AS and the single RS hosting state pension accounts; private state pension accounts are run separately. Is the AS the low- and high-LOA IdP?

(UMA delegation/licensing details on this side elided.) The client application is a special one: a Pensions Dashboard that can aggregate a view of all found pensions. A special Pension Finder Service (not part of UMA) performs the aggregation process.
Scenario: UK Pensions Dashboard

Step 2

Data Subject

Authorization Server Operator

Resource Rights Admin

Resource Server Operator

Pension Finder Service Operator

Requesting Party

Requesting Agent

Resource owner

Delegates

Perm

Authority-to

Delegates

Mgmt-to

Authorization Server Operator

Resource Server Operator

Pension Finder Service Operator

Delegates

Perm

Authority-to

through the Unipass IdP run by Origo for financial advisors, Bob provides high-LOA claims to get access. He may work for himself or a larger firm. Guessing about varying RqP/RqPA relationships.

Alice, now in her shared-with role as “Big Alice”, can now selectively share pension account information to financial advisors from a resource server run by the government that was sourced from the Pension Finder Service. Guessing about the relationships between the services.
Diagrams used in report
(now a bit historical)
Legal roles and artifact interactions

Requesting party

Token (RPT) with permissions

Protection API access token (PAT)

Legal roles and artifact interactions

Key:

- Access granting permissions
- Persisted claims token (PCT)
- Authorization Server Operator
- Resource Server Operator
- Resource Server
- Authorization Server
- Client Operator
- Client
- Requesting Party Agent
- Requesting Party

Legal party name exclusively

Legal party name

UMA technical entity name

UMA party/technical entity name

UMA artifact binding

v.2018-01-15b
Delegation and licensing: RO-centered

Delegates authorization for granting access permissions
- Data Subject
- Resource Owner
- Authorization Server Operator

Delegates management of resources
- Data Subject
- Resource Owner
- Resource Server Operator

Licenses granting access permissions on Resource Owner’s behalf
- Authorization Server Operator
- Resource Server Operator

Limited; Resource Server can only restrict access further, not give more access

UMA artifacts: none

Data Subject
Resource Owner
Authorization Server Operator
Resource Server Operator

Key:
- Legal devices only
- Bound to UMA artifacts

UMA artifacts: Resource Server’s OAuth client credentials, PAT (with Resource Owner context), all request/response messages

v.2018-01-22a
Delegation and licensing: receiving permissions

Licenses receiving access permissions on Resource Owner’s behalf

- Authorization Server Operator
- Client Operator
- Requesting Party

Example message set: Client can revoke RPT to withdraw granted access permissions on Requesting Party’s behalf

UMA artifacts: Client’s OAuth client credentials, RPT (with permissions), claim token, all request/response messages

UMA artifacts: RPT (with permissions), claim token, all request/response messages
Delegation and licensing: RqP-centered

Delegates access seeking

- Requesting Party
- Client Operator

UMA artifacts: claim token, PCT, all request/response messages

Delegates permission to know/persist

- Requesting Party
- Authorization Server Operator

UMA artifacts: PCT, all request/response messages
Earlier group musings
End-to-end licensing relationship

<table>
<thead>
<tr>
<th>Requesting Party</th>
<th>Individual (Bob)</th>
<th>Legal Person (VendorCo)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Owner</td>
<td></td>
<td>Licensee of resource permissions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Individual (Alice)</th>
<th>Individual-to-Vendor</th>
<th>Individual-to-Individual</th>
</tr>
</thead>
<tbody>
<tr>
<td>licensor of resource permissions</td>
<td>Should these be switched? &lt;-&gt; Left is Bob to Alice</td>
<td>JW You are correct - Eve</td>
</tr>
</tbody>
</table>

Sharing Scenario
Sub-licensing intermediaries

Resource Owner

Requesting Party Options

TOS

Client Operator

Individual (self - Alice)

Individual (other - Bob)

Legal Person (VendorCo)

Resource Server Operator

Authorization Server Operator

Individual-to-Self Sharing

Individual-to-Individual Sharing

Individual-to-Vendor Sharing

licensee

sub-licensor

Sharing Scenario
End-to-end licensing relationship (new candidate 2)

<table>
<thead>
<tr>
<th>Requesting Party</th>
<th>Individual (Self)</th>
<th>Individual (other)</th>
<th>Legal Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Owner (Individual)</td>
<td>Individual-to-Self Sharing</td>
<td>Individual-to-Individual Sharing</td>
<td>Individual-to-Vendor Sharing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Client Operator</th>
<th>Resource Server Operator</th>
<th>Authorization Server Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="TOS" /></td>
<td><img src="#" alt="TOS" /></td>
<td><img src="#" alt="TOS" /></td>
</tr>
<tr>
<td><img src="#" alt="TOS" /></td>
<td><img src="#" alt="LIC" /></td>
<td><img src="#" alt="LIC" /></td>
</tr>
<tr>
<td><img src="#" alt="TOS" /></td>
<td><img src="#" alt="LIC" /></td>
<td><img src="#" alt="LIC" /></td>
</tr>
</tbody>
</table>
End-to-end licensing relationship sharing scenarios

<table>
<thead>
<tr>
<th>Requesting Party</th>
<th>Individual (Self)</th>
<th>Individual (other)</th>
<th>Legal Person</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resource Owner (Individual)</strong></td>
<td>Individual-to-Self Sharing</td>
<td>Individual-to-Individual Sharing</td>
<td>Individual-to-Vendor Sharing</td>
</tr>
<tr>
<td><strong>Client Operator</strong></td>
<td><img src="TOS.png" alt="TOS" /></td>
<td><img src="TOS.png" alt="TOS" /></td>
<td><img src="TOS.png" alt="TOS" /></td>
</tr>
<tr>
<td><strong>Resource Server Operator</strong></td>
<td><img src="TOS.png" alt="TOS" /></td>
<td><img src="LIC.png" alt="LIC" /></td>
<td><img src="LIC.png" alt="LIC" /></td>
</tr>
<tr>
<td><strong>Authorization Server Operator</strong></td>
<td><img src="TOS.png" alt="TOS" /></td>
<td><img src="LIC.png" alt="LIC" /></td>
<td><img src="LIC.png" alt="LIC" /></td>
</tr>
</tbody>
</table>

Not sure if this can be incorporated visually, but the arrow of autonomy might be nice. That is to say, who WRITES the TOS or LIC?

If written by RO or rep, autonomy favouring. If by other entity, less so.

JW
How RSO and CO become known to ASO

- Clause text would be supplied for both ToS/PN (non-UMA) and PAT artifacts
- This diagram does not include the RqP-side provisions
- Arrows imply ability for clause text to have the indicated order dependencies
Merging RO-RSO, RO-ASO, and RO-RSO-ASO relationship train tracks

- Clause text would be supplied for both ToS/PN (non-UMA) and PAT artifacts
- This diagram does not include the RqP-side provisions
- Arrows imply ability for clause text to have the indicated order dependencies

Yellow boxes = UMA parties
Merging RqP-CO, RqP-ASO, and RqP-CO-ASO relationship train tracks

- Clause text would be supplied for ToS/PN (non-UMA) artifacts?? Not sure right now
- This diagram does not include the RO-side provisions
- Arrows imply ability for clause text to have the indicated order dependencies

Yellow boxes = UMA parties
RO-RSO-ASO-CO-RqP relationship

- **RO**: OAuth client agreement
- **RSO**: Access contract
- **ASO**: PAT
- **CO**: Agency contract
- **RqP**: RqP TOS/PN
- **PCT (optional)**

Yellow boxes = UMA parties

- **RPT**: UMA artifact, maps to license (legal device)

Sequence

- Arrows imply ability for clause text to have the indicated order dependencies
(Fill in withdrawal/undoing flows)
Example of relationship, legal device, and technical artifact

Legend:
- Red: Pairwise relationship role with greater power
- Green Pairwise relationship role with lesser power
- Blue: Legal device used between them
- Orange: Technical artifact on the UMA wire

The ASO and the RSO have a business contract wherein the ASO, as sub-licensor of resource permissions on behalf of the RO, sub-licenses to the RSO and enables the RSO to sub-license to COs and RqPs by virtue of giving access/giving content.
RqP vs RqPA relationship

**RqPA**
Sharing Scenario B: RqPA was shared with directly by the RO; they are human (Individual). They work for an organization (Legal Person) with which they have an employment agreement (or similar) that is outside the scope of any UMA technical artifacts. Others in the organization might get access by non-UMA methods in downstream fashion, as must be governed by the UMA-enabled license.

**RqP**
(what to call? anything?)
Sharing Scenario B: RqP was shared with directly by the RO; they are an organization (Legal Person). They have humans (Individuals) working for them, with an employment agreement (or similar) that is outside the scope of any UMA technical artifacts, who gets access through non-UMA methods in downstream fashion as governed by the UMA-enabled license.
# Technology/legal stack relationships

<table>
<thead>
<tr>
<th>UMA legal framework</th>
<th>UMA protocol</th>
<th>Framework extension?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consent Receipts?</td>
<td>Some consent tech</td>
<td>PSD2 Consents?</td>
</tr>
<tr>
<td>HL7 Consents?</td>
<td></td>
<td>id-events?</td>
</tr>
<tr>
<td>id-events?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSD2 Consents?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>