UMA Business Model: Mapping graphics
States related to access granting side

<table>
<thead>
<tr>
<th>Single Resource Rights Administrator</th>
<th>Data Subject involved</th>
<th>Data Subject not involved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>State 1: Self-management by Data Subject</td>
<td>State 2: Management under care by Resource Rights Administrator</td>
</tr>
<tr>
<td>Multiple Resource Rights Administrators</td>
<td>State 3: Co-management by Data Subject and Resource Rights Administrator(s)</td>
<td>State 4: Co-management under care by Resource Rights Administrators</td>
</tr>
</tbody>
</table>
State changes related to access granting side

<table>
<thead>
<tr>
<th>Single Resource Rights Administrator</th>
<th>Data Subject involved</th>
<th>Data Subject not involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>State 1: Self-management by Data Subject</td>
<td>State 2: Management under care by Resource Rights Administrator</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multiple Resource Rights Administrators</th>
<th>Data Subject involved</th>
<th>Data Subject not involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>State 3: Co-management by Data Subject and Resource Rights Administrator(s)</td>
<td>State 4: Co-management under care by Resource Rights Administrators</td>
<td></td>
</tr>
</tbody>
</table>

- **Remove Resource Rights Admin: DS revokes RRA delegation**
- **Adjust Data Subject in Resource Rights Admin role**
- **Add Resource Rights Admin: DS delegates control to RRA**
<table>
<thead>
<tr>
<th>States related to requesting side</th>
<th>Individual Requesting Party involved</th>
<th>Individual Requesting Party not involved, or Legal Person Requesting Party</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Requesting Agent</td>
<td>State 5: Self-access by Requesting Party</td>
<td>State 6: Access by Requesting Agent only</td>
</tr>
<tr>
<td>Multiple Requesting Agents</td>
<td>State 7: Co-access by Requesting Party and Requesting Agents</td>
<td>State 8: Access by Requesting Agents only</td>
</tr>
<tr>
<td>State</td>
<td>Single Requesting Agent</td>
<td>Multiple Requesting Agents</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>Requesting Party</td>
<td>State 5: Self-access by Requesting Party</td>
<td>State 7: Co-access by Requesting Party and Requesting Agents</td>
</tr>
<tr>
<td>Requesting Party not involved</td>
<td>State 6: Access by Requesting Agent only</td>
<td>State 8: Co-access by Requesting Party and Requesting Agents</td>
</tr>
</tbody>
</table>

- **Adjust Requesting Agent**
- **Add Requesting Agent**
- **Remove Requesting Agent**
Policy changes based on relationship changes

Alice (RO) --- spouse --- Bob (RqP)
An RO (RRA) and their relationship to an AS and many RS’s
Legal relationships: Persons

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

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

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

**Resource Rights Administrator**

- Is-a

**Authorization Server Operator**

- Is-a

**Resource Server Operator**

- Is-a

**Client Operator**

- Is-a

**Requesting Party**

- Is-a

**Requesting Agent**

- Is-a

**Person**
**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-a
  - 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`
resource owner

authorization server

resource server

requesting party

client

protection API
access token

perm-tokens-pat
resource owner

authorization server

resource server

client

requesting party

requesting party

token

perm-tokens-rpt
A Data Subject may not wish to, or be capable of being, his or her own Resource Rights Administrator (for example, wishing to give power of attorney to someone else or being too young to consent and having a legal guardian as a proxy) and delegates permissions and resource management.

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

A Requesting Party may be an Individual or Legal Person, and a Requesting Agent may also. 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 RqA).
Legal relationships: Devices and artifacts

Making relationships and their changes auditable and machine-readable

1. Resource Rights Administrator
   - Delegates- perm-authority-to
   - Legal devices: ToS, privacy notice (when an Individual)
   - Technical artifacts: Consent management record (outside UMA scope)

2. Authorization Server Operator
   - Delegates-mgmt-to
   - Legal devices: ToS, privacy notice (when an Individual)
   - Technical artifacts: Consent management record (outside UMA scope)

3. Resource Server Operator
   - 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)

Legal devices:
- ToS, privacy notice
- Consent management record

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

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

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

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) (These are all front-channel messages; what are options for auditing?)
Legal relationships: Devices and artifacts

Making relationships and their changes auditable and machine-readable

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?)

Permits-knowing-claims

Delegates-seek-authority-to

client

Authorization Server Operator

requesting party

Client Operator
Legal relationships: Devices and artifacts

Making relationships and their changes auditable and machine-readable

- **Data Subject**
  - Delegates-perm-authority-to
  - Delegates-mgmt-to

- **Legal Person**
  - Delegates-perm-authority-to
  - Delegates-mgmt-to

- **Requesting Party**
  - Delegates-seek-authority-to

- **Resource Rights Administrator**

**Legal devices:** Law or contract

**Technical artifacts:** Outside UMA scope
Legal relationships: One-party/multi-role scenario patterns

In some cases...

- 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.
- A variant where Alice is running a PAS for Johnny.
- Alice has built/is running a “personal data store” for herself.
- A variant where Alice is running a PDS for Johnny.
- 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).
- A variant where this is true of the agent of the ultimate party seeking access. (Included here for completeness but may be too detailed?)
- 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

...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.

Acts-as-a

Authorization Server Operator

Acts-as-a

Client Operator

...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.

Acts-as-a

Authorization Server Operator

Acts-as-a

Identity Provider

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.*
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.)
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?
Scenario: UK Pensions Dashboard

Step 2

Delegates-perm-authority-to

Data Subject

Authorization Server Operator

Resource Rights Admin

resource owner

Delegates-mgmt-to

Resource Server Operator

Delegates-perm-authority-to

Pension Finder Service Operator

Delegates-mgmt-to

Pensions dashboard client

Authorization Server Operator

Unipass high LOA

Requesting Agent

Unipass high LOA

Requesting Party

Unipass high LOA

Ace, 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.

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.
Diagrams used in report
(now a bit historical)
Legal roles and artifact interactions

Legal party name exclusively
Legal party name
UMA technical entity name
UMA party/technical entity name
UMA artifact binding
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
- Authorization Server Operator

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

Data Subject
Resource Owner
Authorization Server Operator
Resource Server Operator

Key:
- Legal devices only
- Bound to UMA artifacts

UMA artifacts:
- none
- Resource Server’s OAuth client credentials, PAT (with Resource Owner context), all request/response messages
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

v.2018-01-22a
Delegation and licensing: RqP-centered

Delegates access seeking

Requesting Party

Client Operator

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

In a Limited Agent role

Delegates permission to know/persist

Requesting Party

Authorization Server Operator

UMA artifacts: PCT, all request/response messages

Key:

Partially bound to UMA artifacts

v.2018-01-22a
Earlier group musings
## End-to-end licensing relationship

<table>
<thead>
<tr>
<th>Requesting Party</th>
<th>Individul (Bob)</th>
<th>Legal Person (VendorCo)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Owner</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Individual-to-Vendor</td>
<td>licensee of resource permissions</td>
</tr>
<tr>
<td>Individual (Alice)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>licensor of resource permissions</td>
<td>Should these be switched? &lt;-&gt; Left is Bob to Alice JW You are correct! - Eve</td>
<td></td>
</tr>
</tbody>
</table>

- JW You are correct! - Eve
Sub-licensing intermediaries

- **Individual (self - Alice)**
  - Requesting Party Options: TOS
  - Resource Owner: Client Operator

- **Individual (other - Bob)**
  - Requesting Party Options: TOS
  - Resource Owner: Client Operator

- **Legal Person (VendorCo)**
  - Requesting Party Options: TOS
  - Resource Owner: Client Operator

- **Resource Server Operator**
- **Authorization Server Operator**

- **Sharing Scenario**
  - Individual-to-Self Sharing
  - Individual-to-Individual Sharing
  - Individual-to-Vendor Sharing
# 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><strong>Resource Owner</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Individual)</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></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOS</td>
<td>TOS</td>
<td></td>
</tr>
<tr>
<td><strong>Resource Server</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator</td>
<td>TOS</td>
<td>LIC</td>
<td></td>
</tr>
<tr>
<td><strong>Authorization</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server Operator</td>
<td>TOS</td>
<td>LIC</td>
<td></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="image" alt="TOS" /></td>
<td><img src="image" alt="TOS" /></td>
<td><img src="image" alt="TOS" /></td>
</tr>
<tr>
<td><strong>Resource Server Operator</strong></td>
<td><img src="image" alt="TOS" /></td>
<td><img src="image" alt="LIC" /></td>
<td><img src="image" alt="LIC" /></td>
</tr>
<tr>
<td><strong>Authorization Server Operator</strong></td>
<td><img src="image" alt="TOS" /></td>
<td><img src="image" alt="LIC" /></td>
<td><img src="image" 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

Yellow boxes = UMA parties

- 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 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
RO-RSO-ASO-CO-RqP relationship

- 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

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.

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>Framework extension?</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMA protocol</td>
<td>Some consent tech</td>
</tr>
</tbody>
</table>

- Consent Receipts?
- HL7 Consents?
- id-events?
- PSD2 Consents?