UMA2 Legal role definitions

Some visualizations presented at IIW 26

(See the companion draft report *A Proposed Licensing Model for User-Managed Access*, available at: `https://kantarainitiative.org/reports-recommendations/`)

Attempt at formal model

Legal relationships: Conventions and terminology

Conventions:

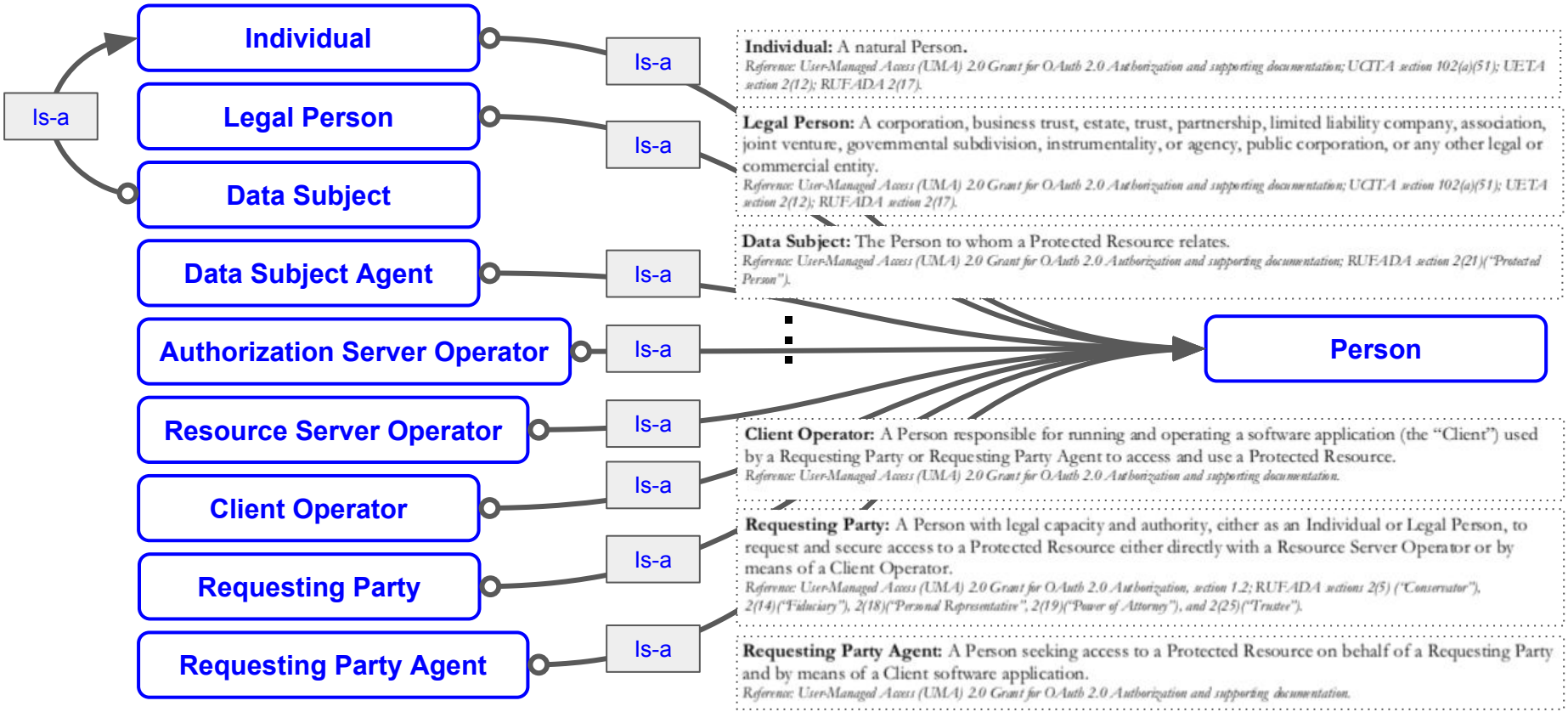
- **Legal (upper capital, blue)** **technical (lowercase, orange)** **issue/question (red)**

Relationship types:

- **Is-a** *(is a kind/species of; for more detail, see the Business Model definitions appendix)*
- **Acts-as-a** *(maps a party defined in the Business Model to a technical entity defined in the specs)*
- Delegates authority for granting and managing access permissions to: **Delegates-perm-authority-to**
- Delegates resource management to: **Delegates-mgmt-to**
- 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**
- **Acts-as-a** (for business scenarios in cases where two roles are served by a single party)

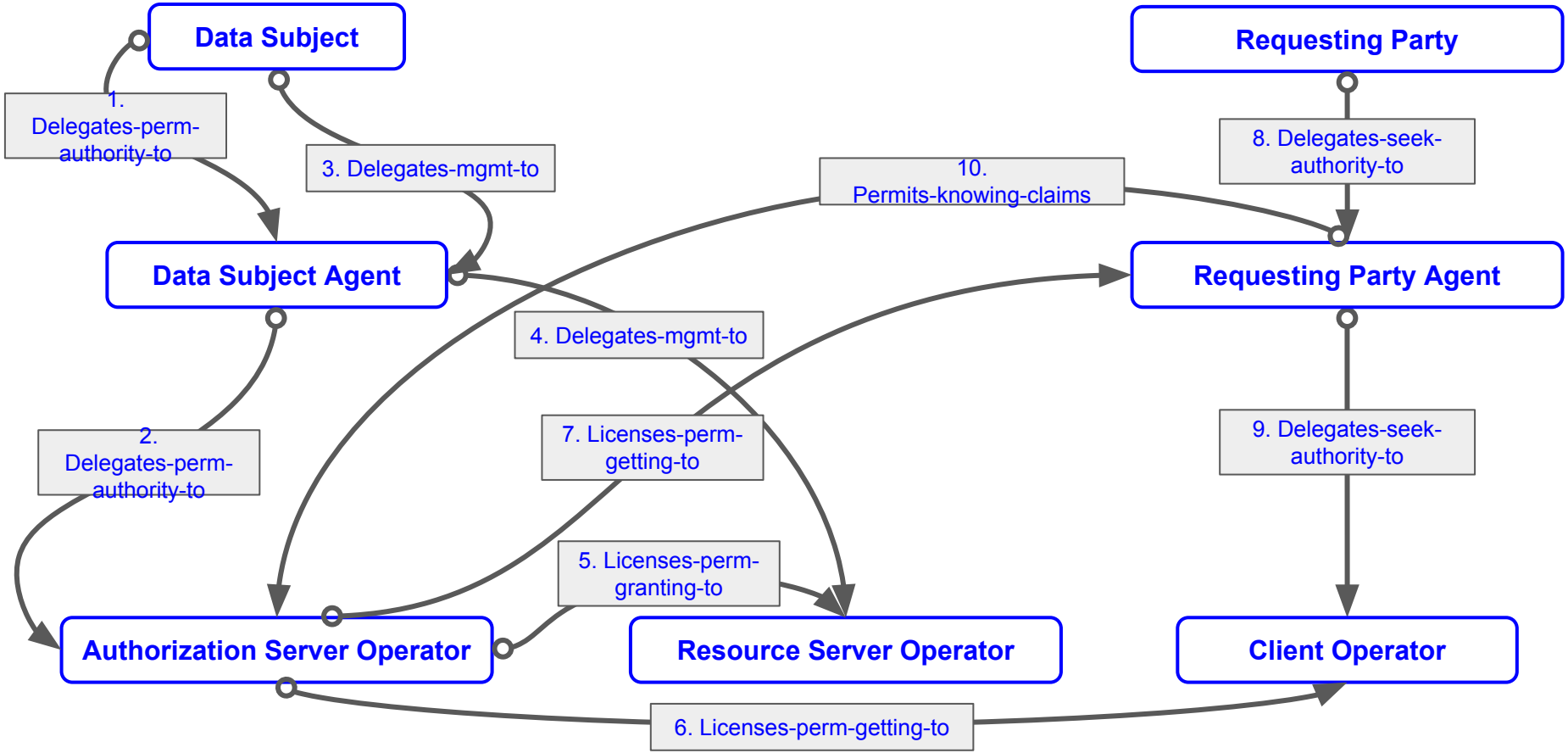
Legal relationships: Persons

Establishes basic party roles



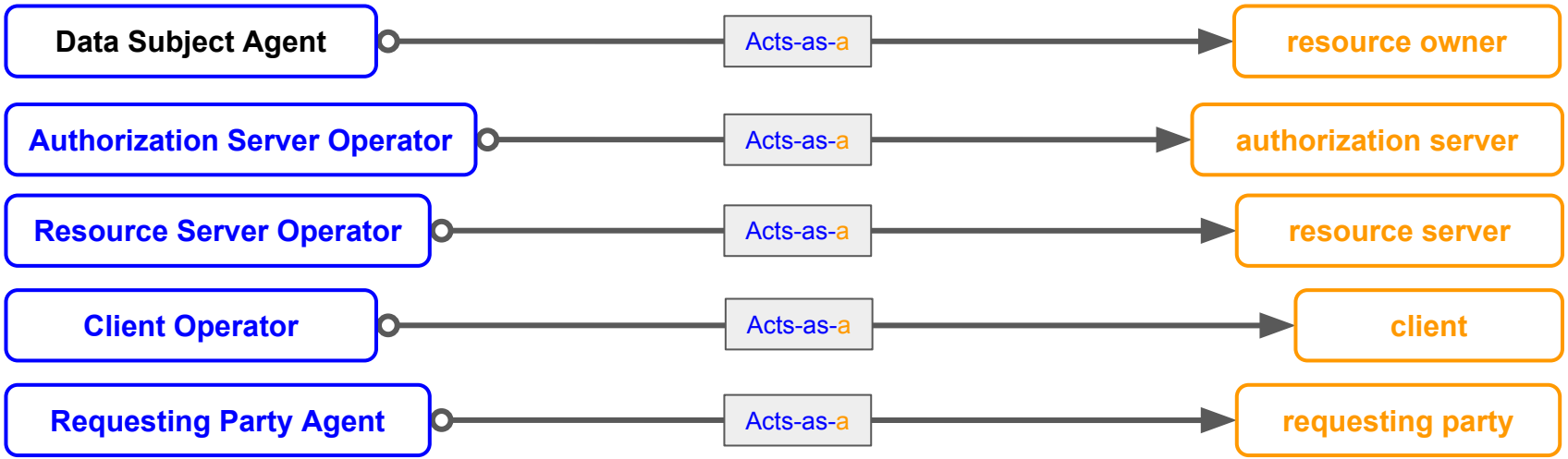
Legal relationships: Delegation and licensing

Establishes how party roles relate to each other in a business sense



Legal relationships: Legal-to-technical role bridges

Establishes how party roles can actually take part in UMA flows



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: Devices and artifacts (1 of 3)

Maps party roles to auditable and machine-readable artifacts

#	Party role	Corresponding entity role	Relationship	Party role	Corresponding entity role	Possible OAuth/ UMA artifacts	Typical legal devices	Comments
1	**Data Subject**	(None)	Delegates-perm-authority-to	**Data Subject Agent**	**resource owner**	(None)	Law or private contract	
2	**Data Subject Agent**	**resource owner**	Delegates-perm-authority-to	**Authorization Server Operator**	**authorization server**	(None)	T/Cs, privacy notice (when DSA is an Individual)	If DSA==ASO, then possibly EULA or nothing (T/Cs: CRs?)
3	**Data Subject**	(None)	Delegates-mgmt-to	**Data Subject Agent**	**resource owner**	(None)	Law or private contract	
4	**Data Subject Agent**	**resource owner**	Delegates-mgmt-to	**Resource Server Operator**	**resource server**	(None)	T/Cs, privacy notice (when DSA is an Individual)	If DSA==RSO, then possibly EULA or nothing (T/Cs: CRs?)
5	**Authorization Server Operator**	**authorization server**	Licenses-perm-granting-to	**Resource Server Operator**	**resource server**	RS OAuth client credentials; PAT with RO context; all AS/RS request/response messages	OAuth client agreement	Agreement is outside/before RO context -- licensing needs to be set up/ prepared there

Legal relationships: Devices and artifacts (2 of 3)

Maps party roles to auditable and machine-readable artifacts

#	Party role	Corresponding entity role	Relationship	Party role	Corresponding entity role	Possible OAuth/ UMA artifacts	Typical legal devices	Comments
6	**Authorization Server Operator**	**authorization server**	Licenses-perm-getting-to	**Client Operator**	**client**	Client OAuth client credentials; RPT with permissions; claim token; all AS/client request/response messages	OAuth client agreement	Agreement is outside/before RqP context -- licensing needs to be set up/ prepared there; important but *non-UMA* artifacts include policies
7	**Authorization Server Operator**	**authorization server**	Licenses-perm-getting-to	**Requesting Party Agent**	**requesting party**	PCT if used, all AS/RqP request/response messages **These are front-channel messages; options for auditing?**	On DS/DSA's behalf, carried through technical artifacts	
8	**Requesting Party**	(None)	Delegates-seek-authority-to	**Requesting Party Agent**	**requesting party**	(None)	Law or private contract	

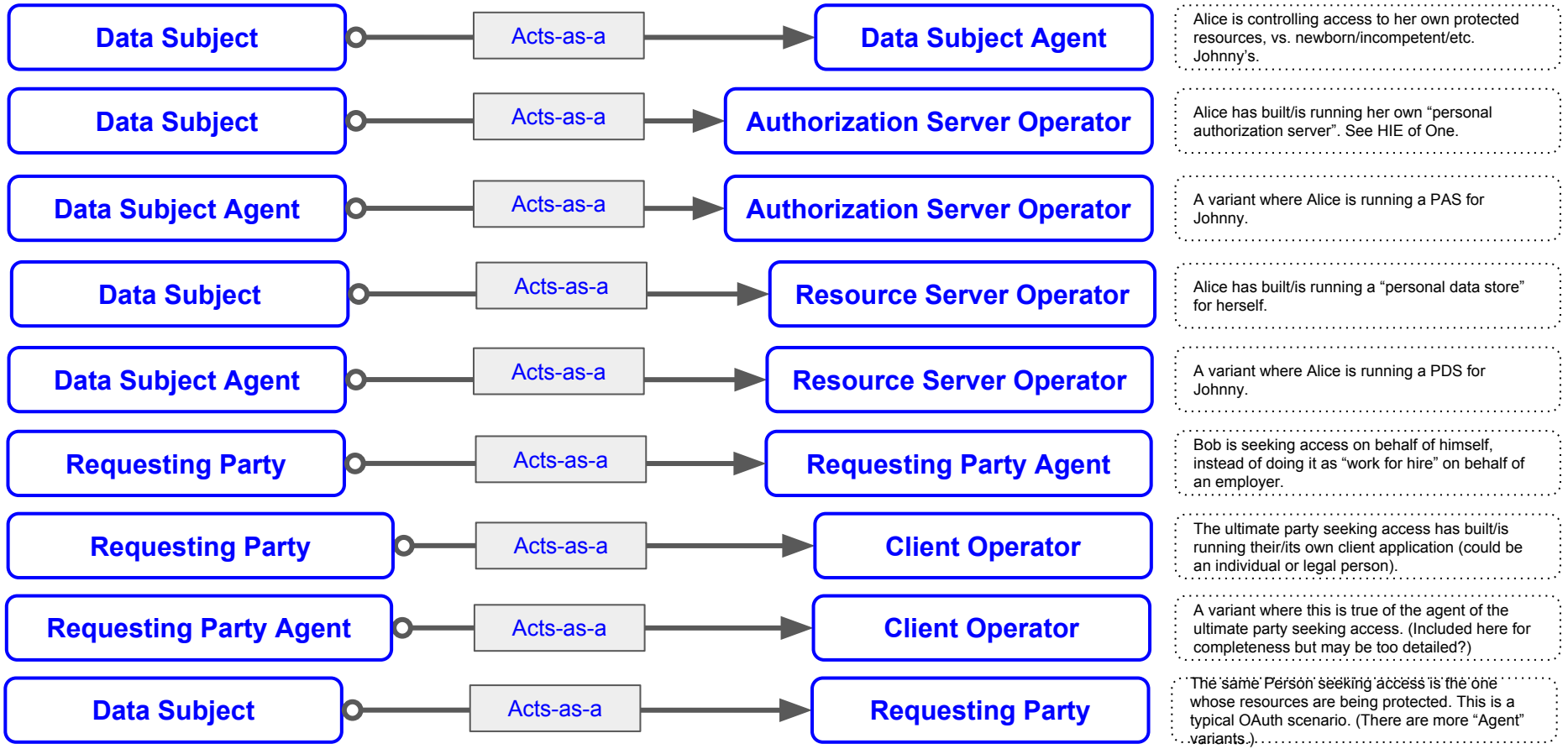
Legal relationships: Devices and artifacts (3 of 3)

Maps party roles to auditable and machine-readable artifacts

#	Party role	Corresponding entity role	Relationship	Party role	Corresponding entity role	Possible OAuth/ UMA artifacts	Typical legal devices	Comments
9	**Requesting Party Agent**	**requesting party**	Delegates-seek-authority-to	**Client Operator**	**client**	Claim token if used, PCT if used, all RqP/client/**AS** request/response messages **Does the AS belong in this list?**	T/Cs, privacy notice (when RqPA is an Individual)	If DSA==CO, then possibly EULA or nothing (T/Cs: CRs?)
10	**Requesting Party Agent**	**requesting party**	Permits-knowing-claims	**Authorization Server Operator**	**authorization server**	PCT if used, all RqP/AS request/response messages	Possibly T/Cs, privacy notice	This is the DS/DSA's ASO (the RO's AS), not (necessarily also) the RqP's AS depending on topology (T/Cs: CRs?)

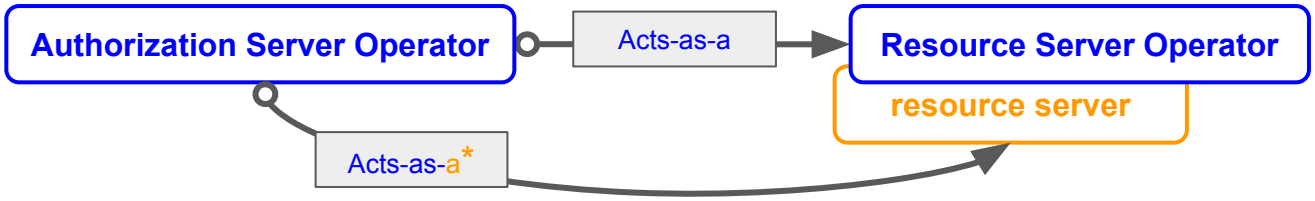
Legal relationships: One-party/multi-role scenario patterns

In some cases...

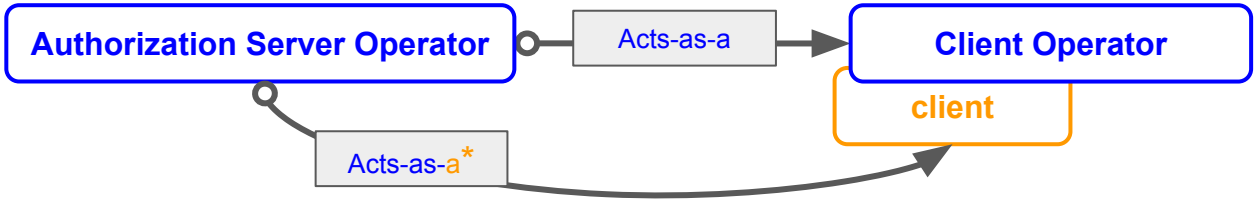


Legal relationships: More scenario patterns

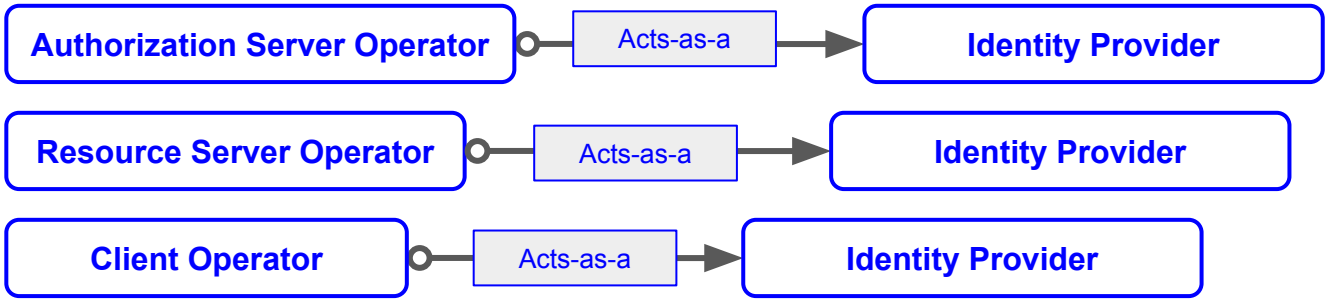
In some cases...



* ...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.



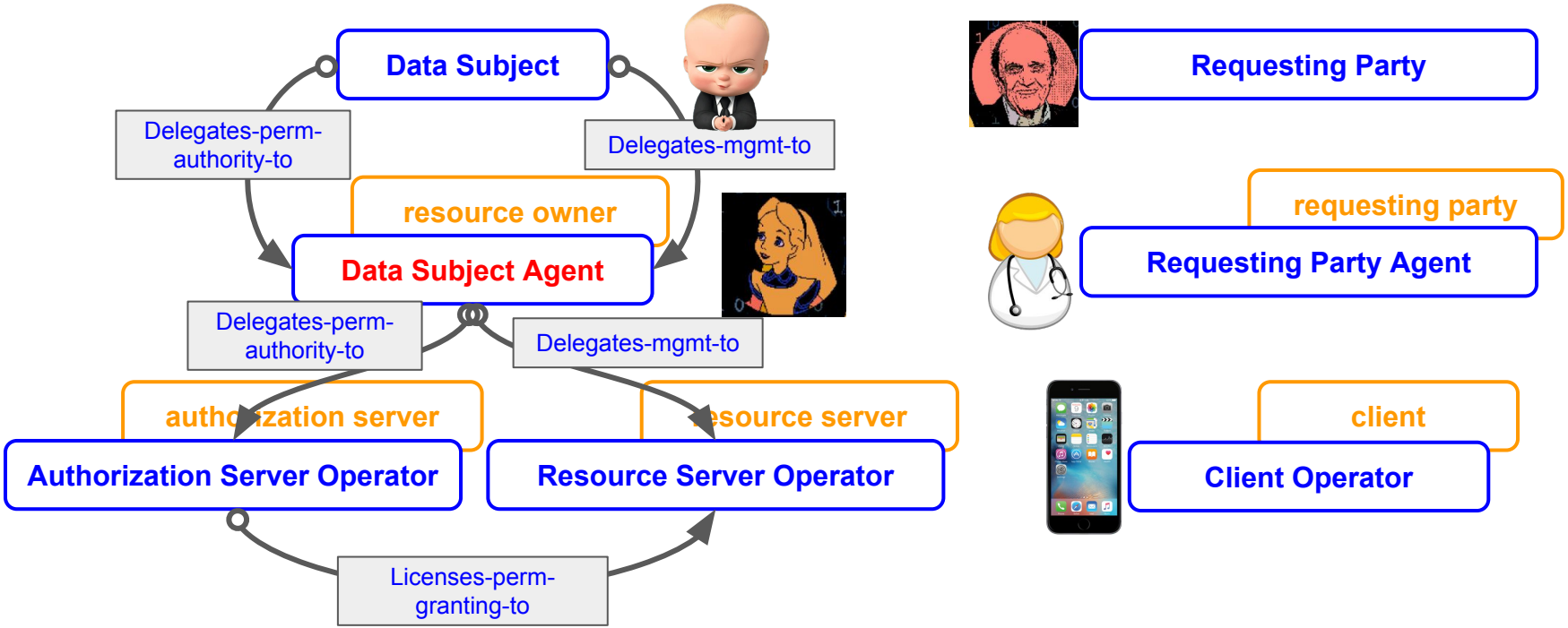
* ...and ASO ***runs all available clients***. This tighter ecosystem (possibly in combination with the above) may still be interested in having the authorization server expose the various UMA interfaces for auditability reasons.



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: Parent-child resource management

Life stage 1

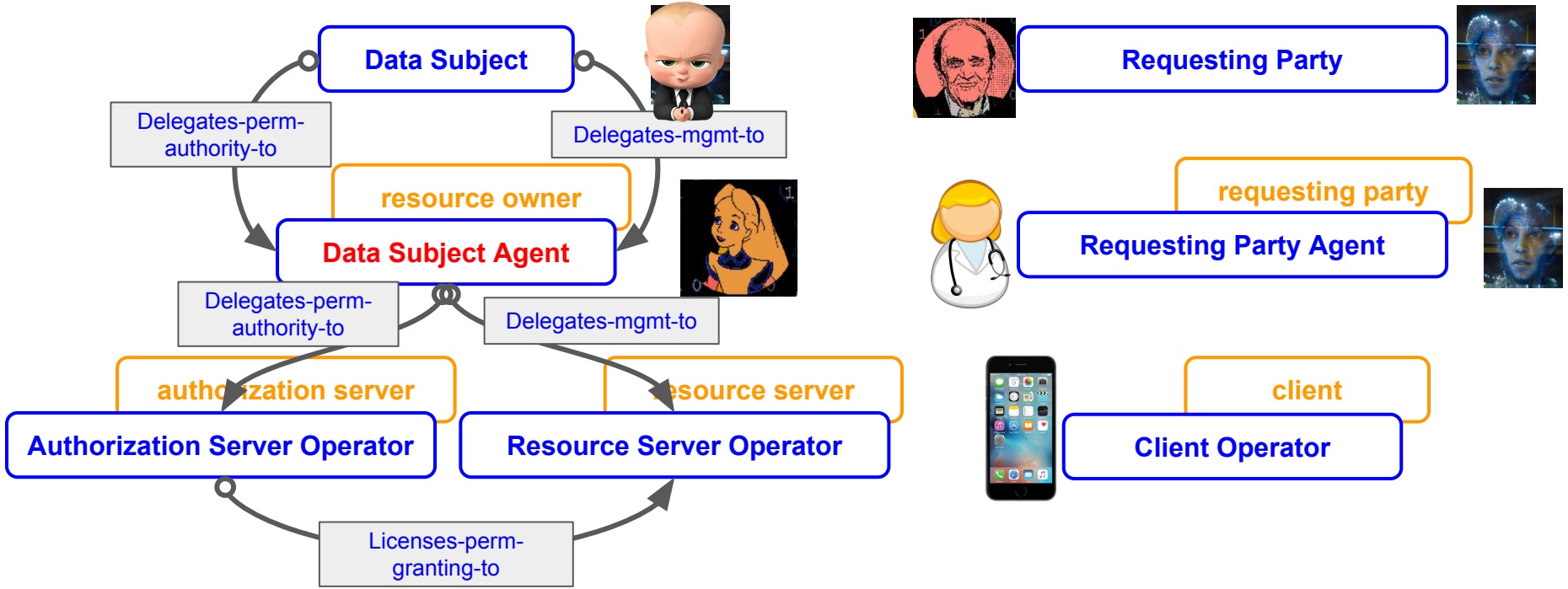


DS is **newborn Johnny**. DSA is **mother Alice**. Delegation from DS to DSA is by law in this case because she is his legal guardian. She manages his protected resources (personal data/digital assets) online and grants access to others on his behalf, for the period that she is his guardian.

(UMA delegation/licensing details on this side elided.) 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.

Scenario: Parent-child resource management

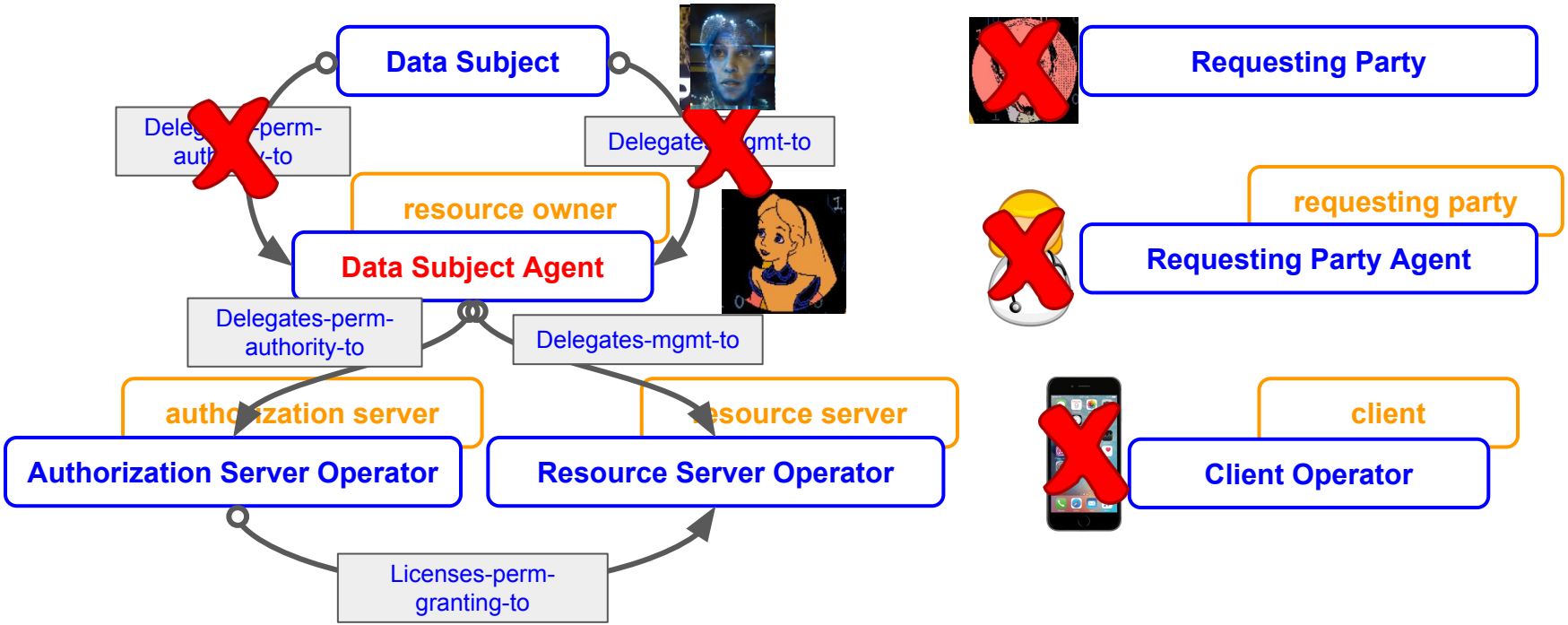
Life stage 2



DS **Johnny** grows old enough to begin using online services. DSA **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, which he could claim and use now.)

Scenario: Parent-child resource management

Life stage 3



DS **Johnny** is old enough to need a legal guardian no longer and may even wish to withdraw his own mother (former DSA) **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.)