Report: UMA Business-Legal Framework and Use Cases

Version: 00
Date: x
UMA Work Group | Kantara Initiative Inc.
Editor: Eve Maler, ForgeRock
Contributors: see Contributors section at end

Abstract: This draft Report outlines a business-legal framework for achieving a broad variety of rights delegation use cases using UMA technology, extending the work of the original draft report A Proposed Licensing Model for User-Managed Access.

Status of This Document: This is an editors' draft produced for consideration by the User-Managed Access Work Group. See the Kantara Initiative Operating Procedures for more information.

Introduction

The connected world requires trust to be built between users and entities sharing personal data across different digital systems. Users need to be assured that only authorized parties can gain access and that these access rights can be monitored, validated, and withdrawn if desired. The User-Managed Access (UMA) standard facilitates the interactions for sharing access by an individual to the personal digital assets they “own” by giving that individual a unified control point for managing access permissions to their personal digital assets no matter where these assets are held.

As a simple example: consider an individual, Alice, whose healthcare records are maintained electronically on her behalf by a local healthcare provider. When applying for life insurance, Alice needs to grant temporary access to her electronic health record (EHR) to her life insurance agent Bob – access she will revoke as soon as the application has been processed, in order to keep her records secure. She may also wish to grant longer-term access to her spouse or her doctor.

A variety of other case studies are discussed on the UMA Case Studies wiki area. UMA enables organizations to address business requirements like these in a secure and standards-based way which can be relied on in the context of growing cybersecurity and regulatory pressures.

The purpose of this Report is to extend the data licensing model discussed in *A Proposed Licensing Model for User-Managed Access*, which introduced how UMA enables the individual to centrally manage access and use rights with respect to personal digital assets. This document will provide a wide variety of example scenarios intended to help business and technical professionals developing and deploying UMA-enabled systems. It will offer a framework for using *permission tokens* to convey machine-readable licenses granting access to personal digital assets. And it will help these professionals understand how technical systems can interact with operational business components in the areas of law, privacy, risk, compliance, security policy, and business policy.

UMA-Enabled Data Sharing: The Typical Use Case

In the typical UMA use case, individuals manage the sharing of their own digital assets with others. In privacy regulatory language, the individual is the *Data Subject*. To enable this type of sharing, the UMA 2.0 specifications define several technical entities.

Let’s use the simple example from the introduction above to introduce these terms. Alice, who has an electronic health record (EHR), is a *resource owner* because her records are in digital form and she manages them herself online. The portal where the EHR lives is called a *resource server*. When applying for life insurance, Alice uses a special service called an *authorization server* to grant temporary EHR access to her life insurance agent Bob. We call Bob a *requesting* *party*. He might use an entirely different application to access her EHR, such as one designed for the insurance profession instead of a patient portal; requesting party apps are called *clients*.

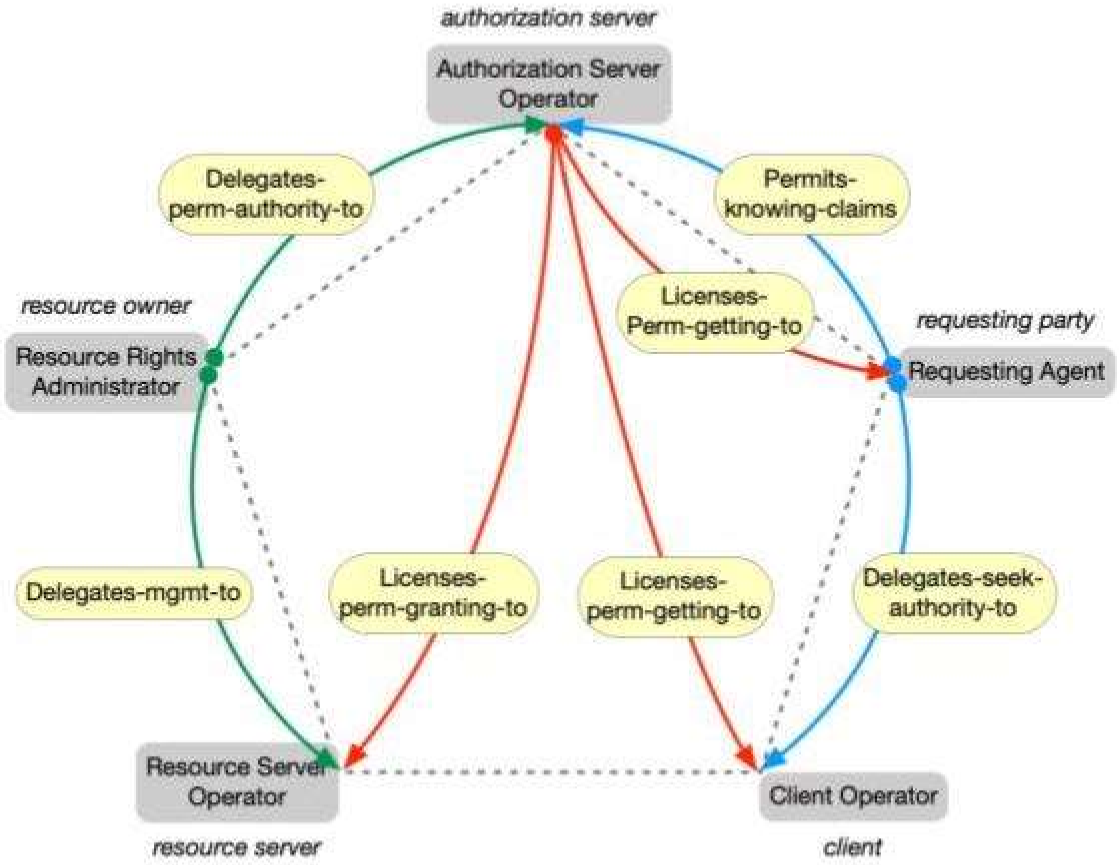
Following is the complete set of formal definitions of the UMA technical entities (whose names are always shown in lowercase):

- **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.
- **client:** An application that is capable of making requests for protected resources with the resource owner's authorization and on the requesting party's behalf.
- **resource server:** A server that hosts resources on a resource owner's behalf and is capable of accepting and responding to requests for protected resources.
- **authorization server:** A server that protects, on a resource owner's behalf, resources hosted at a resource server.

These terms may sound familiar because most of them come from the OAuth 2.0 standard on which UMA is based. OAuth enables Alice to grant permission for a client application that she herself uses to access her desired resource server while avoiding password sharing with that app, a powerful capability. UMA adds several new abilities: Alice can share access with someone entirely different such as Bob instead of just “granting access to herself”. She can decide to share access proactively, before being asked to opt in. She can manage and monitor her access grants at a central authorization server across any number of resource servers run by different organizations. And as a result, UMA enables building new kinds of data and data-sharing ecosystems for the benefit of both individuals and organizations.

These technical roles map closely to business-legal party roles. Let’s explore how this works, using the same use case already seen. Business-legal roles are always shown capitalized. Alice the resource owner serves as the *Resource Rights Administrator* for her own EHR. The resource server hosting that EHR is a *Resource Server Operator*. The authorization server Alice uses to grant EHR access to Bob, her insurance professional, is an *Authorization Server Operator*. We called Bob a requesting party in technical terms, but he’s known in legal party terms as a *Requesting Agent*[1]. The custom client application he uses to access the EHR is software run and operated by a *Client Operator*.

In the figure below, the technical entities are paired with their corresponding legal parties.



Following are the formal definitions for these terms. They refer to other defined terms as well; you can find all such terms in the UMA Business-Legal Framework Use Cases spreadsheet.[2]

- **Resource Rights Administrator:** A Person with legal capacity and authority to act as rights holder, either on behalf of a Data Subject or directly as an Individual or Legal Person, to license access to, sharing, and use of (permissions) relating to a Protected Resource or Informational Rights in a Protected Resource. The Resource Owner is authorized to delegate to an Authorization Server Operator access control, consent, and licensing functions relating to a Protected Resource.
- **Authorization Server Operator**: A Person responsible for running and operating an Authorization Server that controls access and use policies pertaining to Protected Resources on behalf of a Resource Owner; acts as licensing agent for the Resource Owner and may perform these duties by means of an Electronic Agent.
- **Resource Server Operator:** A Person responsible for running and operating a Resource Server that collects, stores, and disseminates Protected Resources: receives licenses from the Authorization Server Operator that provide the RO's permission to give RqP's and CO's access to Protected Resources.
- **Client Operator:** A Person responsible for running and operating a software application (the "Client") used by a Requesting Party or Requesting Agent to access and use a Protected Resource.

- **Requesting Agent:** A Person seeking access to a Protected Resource on behalf of a Requesting Party and by means of a Client software application.

The legal framework requires the following contracts between the legal entities:

1. **Agency Contract**, by which the Resource Rights Administrator delegates or appoints the Authorization Server Operator as licensing agent for access sharing
2. **Access Contracts** between the Authorization Server Operator and each Resource Server Operator, each Requesting Agent, and each Client Operator
3. Machine-readable licenses issued by the Authorization Server:
    a. Protection API access token (PAT) to each Resource Server in a Resource Owner context
    b. Requesting party token (RPT) to each Client on behalf of its Requesting Agent.
    c. Persisted claims token (PCT) to each Client on behalf of its Requesting Agent.

Delegation of Resource Rights Administration

[Use the same example as above, except she’s *representing* little Johnny instead of herself: introduce RRA not-equals DS.]

Having introduced the basics of UMA at both the technical and business-legal levels, we can now explore scenarios where resource rights administration gets more complex. For example, what if Alice must oversee digital assets related to not herself, but her newborn son Johnny? He is too young to manage his own EHR and must rely on his mother as his legal guardian to do so. If she wishes to purchase life insurance for Johnny from Bob, she would be sharing an EHR for someone who is essentially a third-party Data Subject.

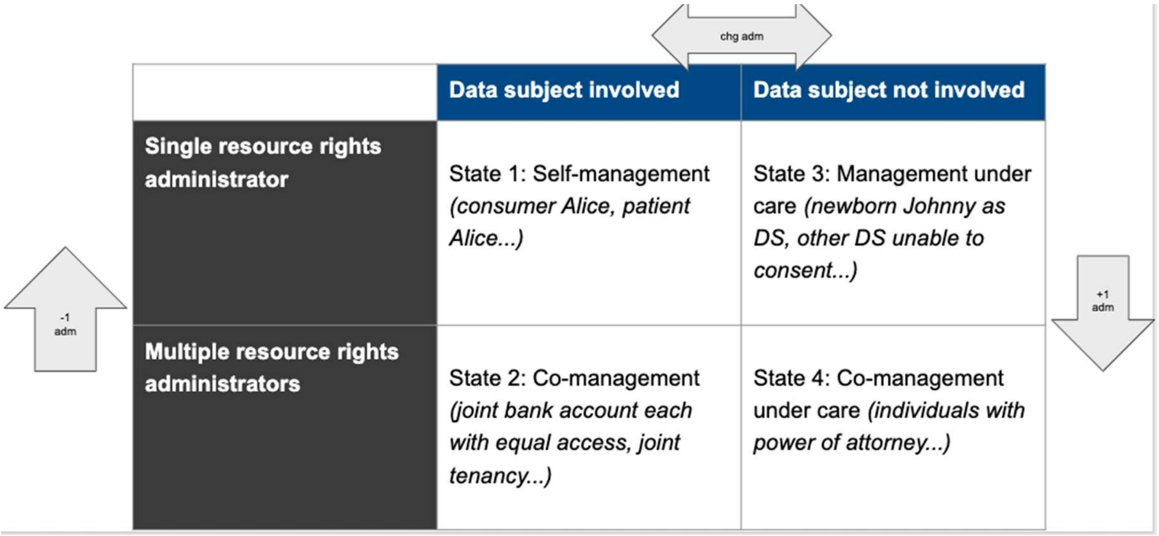
[Introduce table and the four states.]

[Discuss several use cases in healthcare that illustrate the four states.]

[Discuss several bank/financial services use cases that illustrate the four states.]

[Discuss what is required at a B/L level to effect changes from each state to the next. Could vary widely per use case. May require laws, plus identity proofing paperwork, plus self-assertion, etc.]

[Discuss what is required at a T level to effect changes from each state to the next. UMA changes, and also other technical changes. We haven’t worked out a lot of this yet. Permission token flows and consequences.]
]



	Data subject involved	Data subject not involved
Single resource rights administrator	State 1: Self-management *(consumer Alice, patient Alice...)*	State 3: Management under care *(newborn Johnny as DS, other DS unable to consent...)*
Multiple resource rights administrators	State 2: Co-management *(joint bank account each with equal access, joint tenancy...)*	State 4: Co-management under care *(individuals with power of attorney...)*

There may be cases where the Resource Rights Administrator is different than the Data Subject. This is the case in the scenarios with under-age data subjects, who are represented by their guardians, or data subjects under care, e.g., due to mental incapacitation.

There may be multiple 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).

Scenario 1: The data subject is too young to use their digital assets.

Single-carer case:
In this scenario, the data subject is a minor, Johnny. The Resource Rights Administrator is his mother Alice. The delegation of administration from Johnny to Alice is by law as 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.

Multiple-carers case:
There could be multiple carers responsible for Johnny e.g., both parents. [3]

Change Management in Sharing Relationships

[(policies change based on changes in the relationship with RqA?)
Subsections are like part 2's subsections…]

Scenario 1[4]: A data subject, who is a minor and under care, reaches the age of majority.

Data Subject Johnny reaches the age of majority and does not need a legal guardian any longer. Johnny may wish to withdraw his own mother (former Resource Rights Administrator) Alice's access to his resources (personal data).

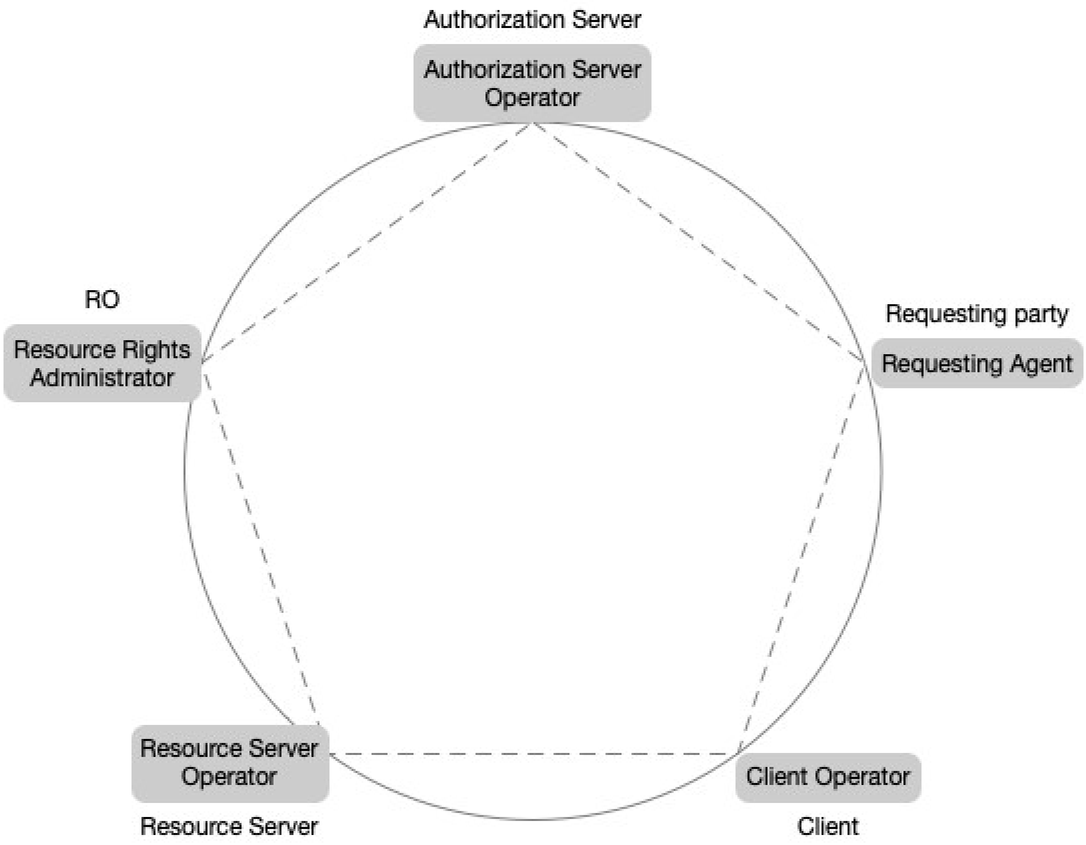
Delegation of Requesting Agent Role

[(e.g., Alice shares her connected car with Bob and he gives his keys to her car to the valet for parking)]

Appendix: Text from Earlier Versions

UMA Technical and Legal Entities

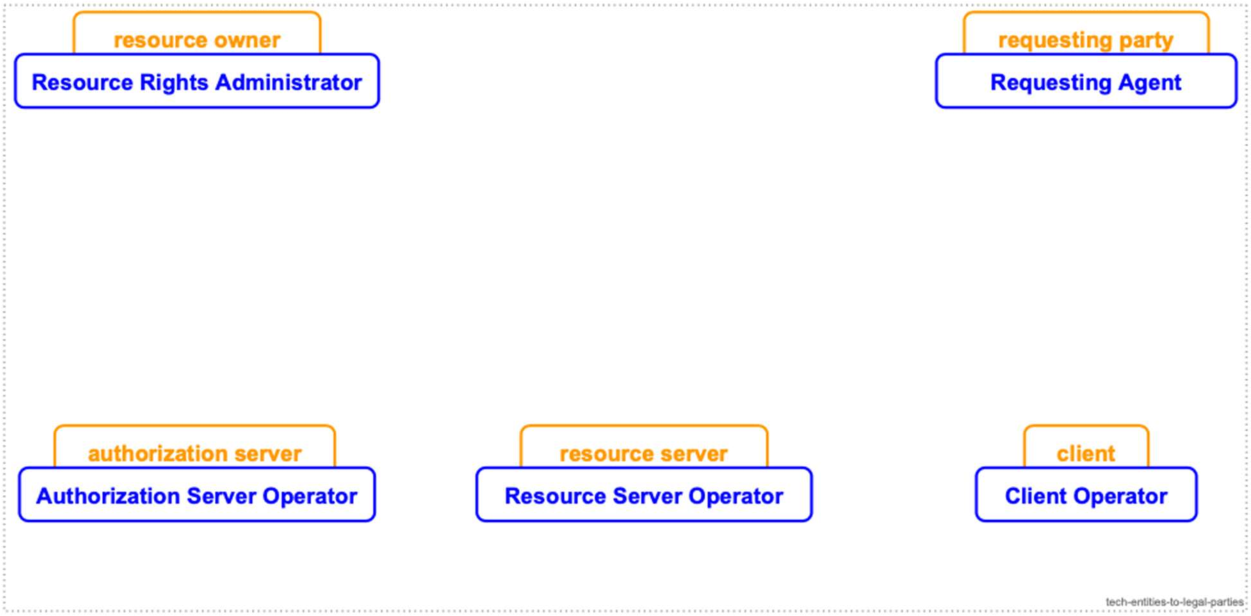
The following figure[5][6][7] shows the technical entities, or actors, in the UMA 2.0 protocol (lowercase and shown on a white background) and how they map to corresponding legal parties (capitalized and shown on a gray background).



The following technical entity definitions quote from the UMA 2.0 Grant specification. While some legal-style language is used here in places, these entities engage in *technical* protocol 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.
- **Client:** An application that is capable of making requests for protected resources with the resource owner's authorization and on the requesting party's behalf.
- **Resource server:** A server that hosts resources on a resource owner's behalf and is capable of accepting and responding to requests for protected resources.
- **Authorization server:** A server that protects, on a resource owner's behalf, resources hosted at a resource server.

The technical entity roles map to legal party roles. Basic mapping is illustrated in the following figure.



The following definitions for these legal party roles are taken from the draft Report *A Proposed Licensing Model for User-Managed Access*.

- **Resource Rights Administrator:** A Person with legal capacity and authority to act as rights holder, either on behalf of a Data Subject or directly as an Individual or Legal Person, to license access to, sharing, and use of (permissions) relating to a Protected Resource or Informational Rights in a Protected Resource. The Resource Rights Administrator is authorized to delegate to an Authorization Server Operator access control, consent, and licensing functions relating to a Protected Resource.
- (@@flesh out)

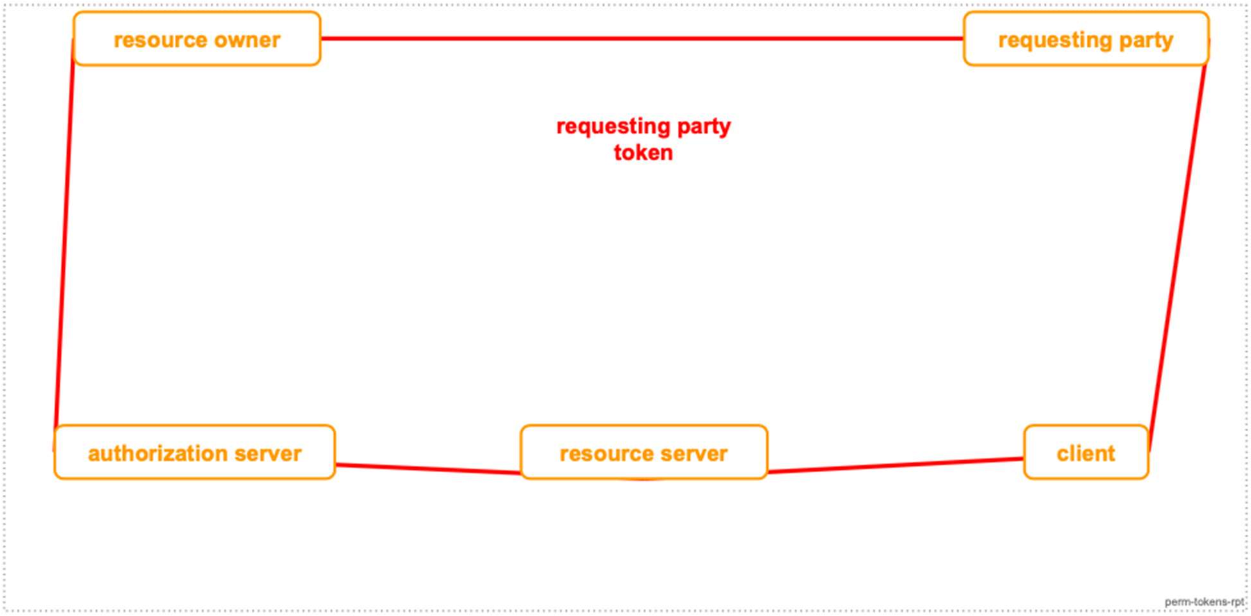
UMA Permission Tokens

Permission tokens are sets of structured information designed to convey security, access control, and/or identity information during cross-entity messaging in a secure and privacy-respecting manner. The authorization server generates three types of tokens in the course of its interactions. Each kind of token creates an association among different entities, and thus among different corresponding legal parties.

The following figure shows how the **protection API access token** (PAT) associates the resource owner, authorization server, and resource server.[8]

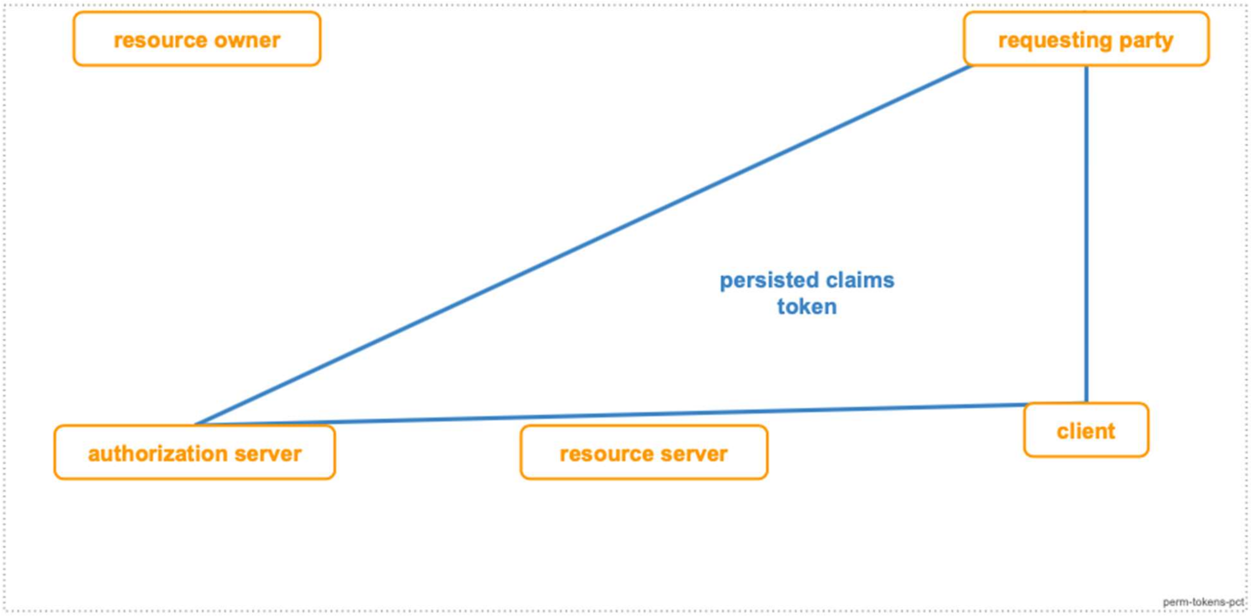
As defined by the UMA 2.0 Federated Authorization specification, the PAT is an OAuth access token "...with the scope `uma_protection`, used by the resource server as a client of the authorization server's protection API. The resource owner involved in the UMA grant is the same entity taking on the role of the resource owner authorizing the issuance of the PAT." The protection API enables multiple resource servers operating in different domains to communicate with a single authorization server operating in yet another domain that acts on behalf of a resource owner.

The following figure shows how the **requesting party token** (RPT) associates all five UMA entities.



The RPT is an OAuth access token used by the UMA grant of OAuth, and thus is the main token involved in UMA.

Finally, the following figure shows how the **persisted claims token** (PCT) associates the requesting party, authorization server, and client.



The PCT is defined in the UMA 2.0 Grant specification as “A correlation handle issued by an authorization server that represents a set of claims collected during one authorization process, available for a client to use in attempting to optimize a future authorization process.” It is optional to produce and use.

Contributors

The following people made significant contributions[9] to this report:

- Andrew Hindle, Hindle Consulting Limited
- Lisa LeVasseur
- Nancy Lush, Lush Group Inc.
- Cigdem Sengul, Nominet
- Timothy S. Reiniger, Esq.

Additional contributors to this specification include the Kantara UMA Work Group participants, a list of whom can be found on the UMA Participant Roster wiki page.