

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 7, 2015

E. Maler, Ed.
ForgeRock
T. Hardjono
MIT
April 5, 2015

Binding Obligations on User-Managed Access (UMA) Participants
draft-maler-oauth-umatrust-03

Abstract

User-Managed Access (UMA) is a profile of OAuth 2.0. UMA defines how resource owners can control protected-resource access by clients operated by arbitrary requesting parties, where the resources reside on any number of resource servers, and where a centralized authorization server governs access based on resource owner policy. This document provides a contractual framework that defines the minimum obligations of parties that operate and use UMA-conforming software programs and services. The goal of this framework is to support end-to-end legal enforceability of the terms and conditions of access sharing relationships between authorizing and requesting sides that use UMA. The audience for this document includes technologists, legal professionals, and operators of UMA-conforming services.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 7, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Sample Use Cases for Sharing Access to Resources	3
1.2.	How to Use the Contractual Framework	4
1.3.	Obligations Not in the Scope of the Contractual Framework	6
2.	Binding Obligations on UMA Participants	7
2.1.	Terminology	7
2.1.1.	Terms	7
2.1.2.	Abbreviations	10
2.2.	Obligations of the Requesting Party	10
2.2.1.	Requesting Party-Authorizing Party: Adhere-to-Terms .	10
2.2.2.	Requesting Party-Authorizing Party: Make-Factual-Representations	11
2.2.3.	Requesting Party-Authorization Server Operator: Supply-Truthful-Claims	11
2.2.4.	Requesting Party-Resource Server Operator: Is-Legitimate-Bearer	12
2.3.	Obligations of the Resource Server Operator	12
2.3.1.	Resource Server Operator-Authorizing Party: Delegate-Protection	12
2.3.2.	Resource Server Operator-Authorization Server Operator: Register-Accurately-and-Timely	12
2.3.3.	Resource Server Operator-Authorization Server Operator: Respect-Permissions	13
2.3.4.	Resource Server Operator-Requesting Party: Give-Accurate-Access	13
2.4.	Obligations of the Authorization Server Operator	14
2.4.1.	Authorization Server Operator-Authorizing Party: Follow-Policies-Accurately-and-Timely	14
2.4.2.	Authorization Server Operator-Resource Server Operator: Follow-Policies-Accurately-and-Timely	14
2.4.3.	Authorization Server Operator-Requesting Party:	

Request-Limited-Claims	15
2.5. Obligations of the Authorizing Party	15
2.5.1. Authorizing Party-Requesting Party: Adhere-to-Terms .	15
2.5.2. Authorizing Party-Authorization Server Operator: Introduce-Resource-Server	15

2.5.3. Authorizing Party-Resource Server Operator: Introduce-Authorization-Server	16
3. Acknowledgments	16
4. References	16
4.1. Normative References	16
4.2. Informative References	17
Appendix A. Document History	17
Authors' Addresses	17

[1.](#) Introduction

User-Managed Access (UMA) [[UMACore](#)] is a profile of OAuth 2.0. UMA defines how resource owners can control protected-resource access by clients operated by arbitrary requesting parties, where the resources reside on any number of resource servers, and where a centralized authorization server governs access based on resource owner policy. This document provides a contractual framework that defines the minimum obligations of parties that operate and use UMA-conforming software programs and services. The goal of this framework is to support end-to-end legal enforceability of the terms and conditions of access sharing relationships between authorizing and requesting sides that use UMA. The audience for this document includes technologists, legal professionals, and operators of UMA-conforming services.

Capitalized terms and abbreviations used in this document are defined in [Section 2.1](#) because they form a normative part of the framework defined in [Section 2](#). Readers are strongly encouraged to review these definitions before reading the rest of the introduction.

[1.1.](#) Sample Use Cases for Sharing Access to Resources

UMA makes possible a loosely coupled end-to-end access sharing relationship between an Authorizing Party and a Requesting Party, with its primary goals being to constrain access according to the Authorizing Party's access policies and to encourage the Requesting

Party to adhere to any obligations it consented to in the authorization process by raising the consequences for doing otherwise. Following are sample use cases that explore the potential differences in these obligations beyond the basic level represented in the contractual framework.

Person-to-self sharing Here, Alice is both the Authorizing Party and the Requesting Party. This use case describes most types of today's OAuth-mediated access, for example, when Alice introduces the Klout service to her Twitter account. She uses both services herself, and wants them to communicate together on her behalf. With UMA, Alice can potentially manage the entire set of such

access connections from a single place rather than from Twitter and other online "home bases" separately. In this circumstance, it's unlikely Alice will want to impose stringent contract terms on herself.

Person-to-person sharing Here, Alice is an Authorizing Party and Bob is a Requesting Party. Today, many Web 2.0 sites offer some level of selective sharing between people, but methods, strengths, and interfaces are inconsistent between sites and people are unable to reuse policies across sites. Alice can share Flickr photos with Bob by adding him to her Flickr friends list or family list or by mailing him a special link to a photo album, or Alice can add Bob as a friend on Facebook. With UMA, Alice can craft authorization policies that let Bob "qualify in" to get access to her photo album and even to other resources she manages at other sites, without her having to be present during this process.

Mediated person-to-organization sharing Here, Alice is an Authorizing Party, the DentalCare company is a Requesting Party, and the company's office assistant Carl is a Requesting Party Agent. Alice wants to give her dentist's office, DentalCare, temporary access to her calendar to make it easier to schedule a series of root canal appointments. Carl might be the actual person acting on behalf of the dental practice who actually asks for and views Alice's calendar. With UMA, Alice can require Carl to prove he is acting on behalf of DentalCare -- for example, demonstrating control of an email address in the dentalcare.com domain -- before seeing her calendar.

Autonomous person-to-organization sharing Here, Alice is an Authorizing Party and the Valley Vehicles car dealership is a Requesting Party. Alice has crafted a "personal request for proposals" because she's in the market for a new car, and she's willing to let car dealerships in her region of the country see her request and make offers to her. With UMA, Valley Vehicles and other dealerships might use Web crawler services to go out and collect requests for proposals, without human helpers, and these services might have to prove in automated fashion that they legitimately represent the right kind of business. Alice can also ensure each dealership agrees to her terms before seeing her request for proposals.

[1.2.](#) How to Use the Contractual Framework

The contractual framework in [Section 2](#) is the normative portion of this document. It is intended to apply to all Subjects that take part in software interactions using services that are declared to be UMA-conforming. It defines the minimum set of obligations that these

Subjects accept. The Subjects can adopt additional obligations, and can further refine or constrain the obligations listed here, but cannot make these minimum obligations less strict.

Each clause takes the following form:

"[_Clause ID_]. When [_protocol interaction takes place_], the [_obligated Subject_] gains an obligation to the [_expecting Subject_] to [_behave towards it in a particular way_]."

The clause ID has the following internal structure:

"[_obligated_-][_expecting_-][_keyword_]"

It uses abbreviations for the obligated and expecting Subjects as follows: RqP for Requesting Party, RSO for Resource Server Operator, ASO for Authorization Server Operator, and AzP for Authorizing Party. Clauses are generally followed by non-normative explanatory comments, which are labeled with "Comments:", and occasionally open issues, which are labeled with "Issues:". The latter are meant to be resolved and removed before final publication.

Specific obligations come as a result of precise protocol interactions, so that at a moment in time, any one Subject may not yet have taken on all of the obligations defined in the contractual framework as belonging to that Subject. By analogy, if Alice were to visit a website that imposes terms of service on the site's users, but it requires users to consent actively by clicking on an "I Agree" button, Alice would take on terms-of-service obligations only after she clicks on the button.

Following are the key UMA interactions that result in obligations, with specific cross-references into the [[UMACore](#)] specification. A non-normative visual correlation of interactions to binding obligations can be found at [[UMAFAQ-swim](#)]. (Lowercase versions of names are references to constructs in the technical specification versus this document.)

- o An authorization server issues a protection API token (PAT) to a resource server: [Section 1.3.1](#).
- o An authorization server issues an authorization API token (AAT) to a client: [Section 1.3.1](#).
- o An authorization server issues a requesting party token (RPT) to a client: [Section 3.4.1](#).

- o An authorization server responds positively to a client's authorization request: [Section 3.4.2](#).
- o A resource server determines the status of an RPT: [Section 3.3](#).
- o A resource server registers a client-requested permission: [Section 3.2](#).
- o A requesting party supplies claims to an authorization server: [Section 3.5](#).
- o A resource server responds to a client's request for access: Sections [3.1.1](#) and [3.1.2](#).
- o A client successfully receives access: [Section 3.1.2](#).

1.3. Obligations Not in the Scope of the Contractual Framework

Of the Subject types defined and discussed in this document, some -- Requesting Party Agents and Client Operators -- have no UMA-dictated obligations, though they might have obligations as part of contractual agreements with other UMA-related Subjects, for example, pairwise contracts or membership in trust frameworks. Additionally, pairs or groups of Subjects that do have obligations imposed by the contractual framework might have additional obligations among themselves beyond those in the framework. Following are some typical examples:

- o When a Resource Owner registers for an account at a Resource Server, the Authorizing Party might gain an obligation to the Resource Server Operator to adhere to the Resource Server Operator's terms of service.
- o When a Client registers with an Authorization Server for OAuth client credentials (for example, through an explicit approval process or through a passive "API-wrap" process), the Client Operator might gain an obligation to the Authorization Server Operator (apart from any particular Requesting Party's usage of that Client) to adhere to the Authorization Server Operator's terms of service for API clients.
- o When a Subject becomes a Requesting Party Agent for a Requesting Party (for example, through an employment agreement), the Requesting Party Agent might gain an obligation to adhere to any agent agreements in place in the Subject's UMA-related interactions performed on behalf of the Requesting Party.

- o When a Requesting Party contracts with a Client Operator to engage in UMA-related interactions on the Requesting Party's behalf, the Client Operator might gain an obligation to adhere to the terms of that contract. For example, a car dealership may contract out to use a cloud service that crawls the Web looking for personal RFPs that meet the dealership's criteria, and want to impose confidentiality requirements.

- o When a Client accesses a protected resource at a Resource Server, the Resource Server Operator might gain an obligation to the Client Operator to be trustworthy as a source of the expected data. For example, in a scenario where the Requesting Party is also the Authorizing Party and is trying to fill in an online loan application through an online financial service (the Client), where the Resource Server Operator provides credit risk data about the Authorizing User, the Client Operator will want to authenticate the Resource Server service in some fashion.

[2. Binding Obligations on UMA Participants](#)

This section constitutes a normative framework that defines the minimum obligations gained by parties that operate and use software programs and services that the operators declare to be UMA-conforming. The framework consists of clauses, where each subsection with content is a clause.

[2.1. Terminology](#)

[2.1.1. Terms](#)

This framework uses the following terms. Where terms are used without capitalization and are not otherwise defined in the [\[UMACore\]](#), they are used in their normal sense.

Individual

A natural person (that is, a human being) with the capacity to take on contractual duties and obligations as a participant in an UMA interaction.

Non-Person Entity (NPE)

A legal person (such as a corporation) with the capacity to take on contractual duties and obligations as a participant in an UMA interaction.

Subject

An Individual or NPE. Subjects play various roles in achieving and seeking user-managed access, and the same Subject might serve in multiple contractual roles.

Claimed adherence of a running software program or service to the requirements of one or more of the roles "authorization server", "resource server", or "client", as defined in [UMACore]. Software components play various roles in participating in the technical interactions necessary to achieve and seek user-managed access, and the same software component might serve in multiple technical roles.

Authorizing Party

A Subject that fills the "resource owner" role as defined in [UMACore], using and configuring software services that variously fill the "authorization server" and "resource server" roles. This Subject is the "user" in "User-Managed Access"; UMA's first priority is to enable Individuals to serve in the Authorizing Party role, though NPEs can serve in this role as well.

Authorization Server

A software service that fills the "authorization server" role as defined in [UMACore].

Authorization Server Operator

A Subject responsible for running and operating an Authorization Server.

Resource Server

A software service that fills the "resource server" role as defined in [UMACore].

Resource Server Operator

A Subject responsible for running and operating a Resource Server.

Client

A software application or service that fills the "client" role as defined in [UMACore].

Client Operator

A Subject responsible for running and operating a Client.

Requesting Party

A Subject that uses a Client to seek access to a protected resource. This Subject may be an Individual or an NPE. The Requesting Party and the Authorizing Party may be the same Subject or different Subjects.

Requesting Party Agent

A Subject using a Client to seek access to a protected resource on behalf of a Requesting Party. Typically this Subject is an Individual acting on behalf of an NPE.

Comments: The [\[UETA\]](#) defines two terms that are particularly relevant to understanding the interactions among UMA participants:

- o "'Automated transaction' means a transaction conducted or performed, in whole or in part, by electronic means or electronic records, in which the acts or records of one or both parties are not reviewed by an individual in the ordinary course in forming a contract, performing under an existing contract, or fulfilling an obligation required by the transaction."
- o "'Electronic agent' means a computer program or an electronic or other automated means used independently to initiate an action or respond to electronic records or performances in whole or in part, without review or action by an individual."

Where a Client is used by a human Requesting Party or a human Requesting Party Agent, at times human-computer interaction (HCI) will be required, but otherwise the access-attempt transaction is likely to be fully automatic from the perspective of the "requesting side". Furthermore, where the Authorizing Party and the Requesting Party are the same natural person, or where the Authorizing Party has set a policy that requires real-time approval through some out-of-band method, this person can expect to engage in HCI. Otherwise the access-attempt transaction is likely to be fully automatic from the perspective of the "authorizing side" because the access attempt is made without any requirement for the Authorizing Party to be present at run time.

The National Strategy for Trusted Identities in Cyberspace [\[NSTIC\]](#) defines some terms similar to those defined here:

- o "An individual is a person engaged in an online transaction. Individuals are the first priority of the Strategy."
- o "A non-person entity (NPE) may also require authentication in the Identity Ecosystem. NPEs can be organizations, hardware, networks, software, or services and are treated much like individuals within the Identity Ecosystem. NPEs may engage in or support a transaction."
- o "The subject of a transaction may be an individual or an NPE."

UMA shares with NSTIC a priority to enable and empower individual people in the context of their online interactions. Note that this

framework uses the terms Individual, NPE, and Subject exclusively for parties that have the capacity to take on contractual obligations, distinguishing them "from hardware, networks, software, or services", which do not have this capacity.

[2.1.2.](#) Abbreviations

This framework uses the following abbreviations.

UMA User-Managed Access, the interoperability protocol defined by in [\[UMACore\]](#) and the other specifications it includes normatively by reference.

API Application programming interface.

PAT Protection API token, as defined in [\[UMACore\]](#).

AAT Authorization API token, as defined in [\[UMACore\]](#).

RPT Requesting party token, as defined in [\[UMACore\]](#).

Comments: Tokens are critical to managing authorization and auditing of resource access. [Section 1.3](#) is recommended reading for understanding what the various tokens represent and how they are issued and used. The RPT, in particular, has a definition that can vary depending on the RPT profile in use; thus, any obligations in this framework that depend on an RPT profile specify it by name.

[2.2.](#) Obligations of the Requesting Party

[2.2.1.](#) Requesting Party-Authorizing Party: Adhere-to-Terms

When the Client successfully gains access from a Resource Server to a protected resource by wielding a valid "bearer" RPT associated with at least one currently valid permission for the type of access sought, the Requesting Party using that Client gains an obligation to the Authorizing Party to adhere to any terms it agreed to in order to gain the permission.

Comments: This key obligation enables the end-to-end access authorization agreement that UMA exists to forge. At a previous stage, the Requesting Party asked for a relevant permission from the Authorization Server and might have had to provide claims of a promissory nature. Accepting access to the protected resource binds the Requesting Party to any terms it agreed to using the claims mechanism, for example, agreeing only to read the resource rather than modifying it, or forbearing from selling the resource data to someone else.

Issues: Note that the obligation goes into effect the first time a Client gains access under the power of a "currently valid permission". If there was more than one valid permission attached to different sets of promises, if a secure record was not kept by the Resource Server and/or Authorization Server of which permission was used for granting access, ambiguity is introduced. Defining and using RPT profiles other than the "bearer" profile might lessen the potential ambiguity.

[2.2.2.](#) Requesting Party-Authorizing Party: Make-Factual-Representations

When the Requesting Party provides, or facilitates the sourcing of, claims to an Authorization Server in a claims-gathering flow, the Requesting Party gains an obligation to the Authorizing Party to stand behind any factual representations it makes in order to gain the permission, to the best of its knowledge at the time it makes them.

Comments: This obligation is gained during the providing of actual claims, rather than at the time of AAT issuance or protected resource access, because factual claims might age and expire. Where the Requesting Party supplies or sources claims in a manner that can be verified by the Authorization Server, the risk imposed by this need for "trust" can be reduced. Note that UMA defines an optional OpenID Connect claim profile that provides one way to collect trusted claims from third-party claim providers.

[2.2.3.](#) Requesting Party-Authorization Server Operator: Supply-Truthful-Claims

When the Authorization Server issues an AAT to a Client and for as long as the AAT is valid, the Requesting Party using that Client

gains an obligation to the Authorization Server Operator to supply or facilitate access to truthful claims required for access authorization at this AM, when it chooses to supply them, to the best of its knowledge at the time it supplies them.

Comments: At a later stage, the Requesting Party might be asked to provide claims to support authorization processes at this Authorization Server for accessing `_all_` resources protected by this Authorization Server, managed by any Authorizing Parties, at any Resource Servers. The Requesting Party's responsibility to act in good faith in interacting with this Authorization Server begins now because factual claims it supplies could be reused for more than one access-sharing relationship. This obligation can be removed through AAT revocation.

[2.2.4.](#) Requesting Party-Resource Server Operator: Is-Legitimate-Bearer

When the Authorization Server issues an RPT to a Client and for as long as the RPT is valid, the Requesting Party using that Client gains an obligation to the Resource Server Operator to represent the legitimate bearer of the RPT or its authorized representative, and not to allow others to impersonate the Requesting Party.

Comments: In the case where the "bearer" RPT profile or any other bearer-style RPT profile is used, the token may not be bound in any technically confirmable way to the specific client and requesting party it applies to. Defining and using different UMA token profiles can mitigate the risk of failure or malice on the Requesting Party's part. The "authorized representative" phrase is intended to clear the way for token-chaining profiles or similar.

[2.3.](#) Obligations of the Resource Server Operator

[2.3.1.](#) Resource Server Operator-Authorizing Party: Delegate-Protection

When the Authorization Server issues a PAT to a Resource Server and as long as the PAT is valid, the Resource Server Operator gains an obligation to the Authorizing Party to delegate protection services to the Authorization Server Operator for the set of protectable resources for which it represents this capability to the Authorizing

Party, and to respect the authorization data that the Authorization Server has associated with an RPT when the Resource Server subsequently allows or disallows access by the Client that presented that RPT.

Comments: Once the Authorization Server Operator becomes the Authorizing Party's authorization proxy, it begins relying on the Resource Server Operator in other, more specific ways. The Resource Server has the opportunity to inspect AM-issued permissions or take other actions that delegate protection responsibility to the Authorization Server at a later stage, but its responsibility for respecting them begins now. The specific protection services made available to the Resource Server by the Authorization Server differ depending on the RPT profile in use. This obligation can be removed through PAT revocation.

[2.3.2.](#) Resource Server Operator-Authorization Server Operator: Register-Accurately-and-Timely

When the Authorization Server issues a PAT to the Resource Server and as long as the PAT is valid, the Resource Server Operator gains an obligation to the Authorization Server Operator to register resource

set descriptions accurately and timely according to the Authorizing Party's expressed instructions for protection.

Comments: At a later stage, the Resource Server has the opportunity to register resource sets, but its responsibility for performing this task begins now. The Resource Server Operator may have contracted with the Authorizing Party for service-level agreements to respond specifically to timeliness needs and so on. This obligation can be removed through PAT revocation.

[2.3.3.](#) Resource Server Operator-Authorization Server Operator: Respect-Permissions

When the Resource Server successfully introspects a "bearer" RPT, the Resource Server Operator gains an obligation to the Authorization Server Operator to respect the permissions that the Authorization Server has associated with the RPT when the Resource Server subsequently allows or disallows access by the Client that presented

that RPT.

Comments: The Resource Server Operator, as a Subject that is otherwise potentially not obligated to the Authorization Server Operator, carries a great deal of responsibility here not to allow access where the Authorization Server has not granted permission and to make every effort to grant access where the Authorization Server has granted permission. Its interpretation of scopes and permissions is not directly auditable by the Authorization Server. Further, issues can arise from the skew between permission validity periods and actual access. Defining and using different RPT profiles can mitigate the risk of failure or malice on the Resource Server Operator's part.

[2.3.4.](#) Resource Server Operator-Requesting Party: Give-Accurate-Access

When the Resource Server responds in any fashion to a Client's access request, the Resource Server Operator gains an obligation to the Requesting Party to give accurate access to the protected resource according to whether the Requesting Party has permission to do so.

Comments: The Resource Server Operator, as a Subject that is otherwise potentially not obligated to the Authorization Server Operator, carries a great deal of responsibility here to make every effort to grant access where the Authorization Server has associated authorization data to guide access. Its interpretation of scopes and permissions, particularly in the case where the RPT presented by the Client uses the "bearer" RPT profile, is not entirely auditable by the requester or Authorization Server. Further, issues can arise from the skew between permission validity periods and actual access.

Defining and using different RPT profiles can mitigate the risk of failure on the Resource Server Operator's part.

Issues: The relying party traditionally always has the right of refusal. The resource server may have additional authorization context available only to it that suggests it should not grant access, for example. Should this obligation be struck?

[2.4.](#) Obligations of the Authorization Server Operator

[2.4.1.](#) Authorization Server Operator-Authorizing Party: Follow-

Policies-Accurately-and-Timely

When the Authorization Server issues a PAT to the Resource Server and as long as the PAT is valid, the Authorization Server Operator gains an obligation to the Authorizing Party to adhere to the Authorizing Party's policies accurately and timely in granting permissions.

Comments: At a later stage, the Authorization Server will require the Resource Server to present the PAT whenever it uses the Authorization Server's protection API on behalf of this Authorizing Party. The Authorization Server Operator may have contracted with the Authorizing Party for service-level agreements to respond specifically to timeliness needs and so on. This obligation can be removed through PAT revocation.

[2.4.2.](#) Authorization Server Operator-Resource Server Operator: Follow-Policies-Accurately-and-Timely

When the Resource Server registers a requested permission at the Authorization Server, the Authorization Server Operator gains an obligation to the Resource Server Operator to adhere to the Authorizing Party's authorization policies accurately and timely in associating authorization data with RPTs presented with the registered permission's ticket.

Comments: At a later stage, when a Client approaches the Authorization Server presenting an RPT and a permission ticket, the Authorization Server matches Authorizing Party policies to the requested permission to drive any requests for claims and ultimate authorization processes, but its responsibility for performing this task begins now.

[2.4.3.](#) Authorization Server Operator-Requesting Party: Request-Limited-Claims

When the Authorization Server issues an AAT to a Client and as long

as the AAT is valid, the Authorization Server Operator gains an obligation to the Requesting Party to request only claims that support the purpose of satisfying an Authorizing Party's policy.

Comments: At a later stage, the Authorization Server might ask the Requesting Party to provide claims for specific permission purposes at multiple Resource Servers and/or for multiple Authorizing Parties, but its responsibility begins now. This obligation can be removed through AAT revocation.

[2.5.](#) Obligations of the Authorizing Party

[2.5.1.](#) Authorizing Party-Requesting Party: Adhere-to-Terms

When the Authorization Server responds positively to a Client's request for authorization, the Authorizing Party gains an obligation to the Requesting Party using that Client to adhere to the terms offered to and accepted by the Requesting Party in the form of requests for claims driven by the Authorizing Party's policy at the Authorization Server.

Comments: For example, the Authorizing User cannot subsequently protest or sue the Requesting Party for resale of the user's data if this was allowed by the terms of access authorization.

[2.5.2.](#) Authorizing Party-Authorization Server Operator: Introduce-Resource-Server

When the Authorization Server issues a PAT to a Resource Server and as long as the PAT is valid, the Authorizing Party gains an obligation to the Authorization Server Operator to introduce the desired Resource Server to this Authorization Server in outsourcing protection of this Resource Server's resources.

Comments: How the Resource Server learned of the Authorization Server's location is out of band for UMA; it is the Authorizing Party's responsibility to check that it has been redirected to an acceptable Authorization Server before the Authorization Server successfully issues the PAT. This obligation can be removed through PAT revocation.

[2.5.3.](#) Authorizing Party-Resource Server Operator: Introduce-Authorization-Server

When the Authorization Server issues a PAT to a Resource Server and as long as the PAT is valid, the Authorizing Party gains an obligation to the Resource Server Operator to introduce the desired Authorization Server to this Resource Server in outsourcing protection of this Resource Server's resources.

Comments: Once the Authorization Server Operator becomes the Authorizing Party's authorization proxy, the Resource Server Operator begins relying on it in other, more specific ways. How the Authorizing Party indicated the desired Authorization Server to the host is out of band for UMA; it is the Authorizing Party's responsibility to check that it has been redirected to an acceptable Authorization Server before the Authorization Server successfully issues the PAT. This obligation can be removed through PAT revocation.

[3.](#) Acknowledgments

The current editor of this document is Eve Maler of XMLgrl.com. The following people are co-authors:

- o Domenico Catalano, Oracle Corp.
- o Kevin Cox, Edentiti
- o Sal D'Agostino, IDmachines
- o Susan Morrow, Avoco Secure
- o Dazza Greenwood, Civics.com

Additional contributors to this document include the Kantara UMA Work Group participants, a list of whom can be found at [[UMAnitarians](#)]. The co-authors and contributors thank Scott David, Dazza Greenwood, and Tom Smedinghoff for offering their legal expertise and insight in the preparation of this document.

[4.](#) References

[4.1.](#) Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

Internet-Draft

UMA Binding Obligations

April 2015

[UMAcore] Hardjono, T., "User-Managed Access (UMA) Profile of OAuth 2.0", December 2012,
<<http://tools.ietf.org/html/draft-hardjono-oauth-umacore>>.

4.2. Informative References

[NSTIC] US Federal Government, "National Strategy for Trusted Identities in Cyberspace", April 2011,
<http://www.whitehouse.gov/sites/default/files/rss_viewer/NSTICstrategy_041511.pdf>.

[UETA] Smedinghoff, T., "Uniform Electronic Transactions Act", 1999,
<<http://www.law.upenn.edu/bll/archives/ulc/fnact99/1990s/ueta99.htm>>.

[UMAFAQ-swim] US Federal Government, "UMA Frequently Asked Questions: What is the UMA protocol flow?", January 2013,
<<http://kantarainitiative.org/confluence/display/uma/UMA+FAQ#UMAFAQ-WhatistheUMAprocolflow>>.

[UMAnitarians] Maler, E., "UMA Participant Roster", 2012,
<<http://kantarainitiative.org/confluence/display/uma/Participant+Roster>>.

Appendix A. Document History

NOTE: To be removed by RFC editor before publication as an RFC.

Authors' Addresses

Eve Maler (editor)
ForgeRock

Email: eve.maler@forgerock.com

Thomas Hardjono

MIT

Email: hardjono@mit.edu