Binding Obligations on User-Managed Access (UMA)
Participants
draft-maler-uma-trust-00c

Abstract

The User-Managed Access (UMA) core protocol provides a method for users to control access to their protected resources, residing on any number of host sites, through an authorization manager that governs access decisions based on user 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 overall goal of this framework is to support legal enforceability of access-sharing agreements made between the authorizing and requesting sides using 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 January 25, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 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

The User-Managed Access (UMA) core protocol [UMAcore] provides a method for users to control access to their protected resources, residing on any number of host sites, through an authorization manager that governs access decisions based on user 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 overall goal of this framework is to support legal enforceability of access-sharing agreements made between the authorizing and requesting sides using UMA. The audience for this document includes technologists, legal professionals, and operators of UMA-conforming services.

Capitalized terms and abbreviations in this document are defined in Section 2.1.

1.1. Sample Use Cases for Sharing Access to Resources

UMA makes possible an 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 an Authorizing Party and a 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 this control, 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 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 that are declared to be UMA-conforming. It define the minimum set of obligations that these Subjects accept. The Subjects can adopt additional obligations, and 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 in a particular way]."

The clause ID uses the abbreviation T for terminology, R for Requesting Party, H for Host Operator, AM for AM Operator, and AP for Authorizing Party respectively. Clauses are sometimes followed by non-normative explanatory comments, which are labeled with "Comments:", and open issues, which are labeled with "Issues:" and are meant to be resolved and removed before final publication.

The various Subjects take on binding obligations by interacting with the others using software programs or services that are declared by their operators to be UMA-conforming. The 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 pushes the button.

Following are the key UMA interactions that result in obligations, along with references to the
relevant sections in [UMAcore] that define these interactions:

- A Host responds to a Requester's access request: Section 3.1.
- An AM issues a protection API token (PAT) to a Host: Section 2.3.
- An AM issues an authorization API token (AAT) to a Requester: Section 3.4.3.
- An AM issues a requester permission token (RPT) to a Requester: Section 3.4.4.
- An AM responds positively to a Requester's permission request: Section 3.4.5.
- A Host checks the status of an RPT: Section 3.3.
- A Host registers a permission on behalf of a Requester: Section 3.2.
- A Requester supplies claims to an AM: Section 3.5.
- A Requester successfully receives access: Section 3.1.3.

1.3. Obligations Not in the Scope of the Contractual Framework

Of the Subject types discussed in this document, some -- Requesting Party Agents and Requester 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:

- When an Authorizing Party registers for an account at a Host, the Authorizing Party might gain an obligation to the Host Operator to adhere to the Host Operator's terms of service.
- When a Requester registers with an AM for OAuth client credentials (for example, through an explicit approval process or through a passive "API-wrap" process), the Requester Operator might gain an obligation to the AM Operator (apart from any particular Requesting Party's usage of that Requester) to adhere to the AM Operator's terms of service for API clients.
- 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.
- When a Requesting Party contracts with a Requester Operator to engage in UMA-related interactions on the Requesting Party's behalf, the Requester 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.
- When a Requester accesses a protected resource at a Host, the Host Operator might gain an obligation to the Requester 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 Requester), where the Host Operator provides credit risk data about the Authorizing User, the Requester Operator will want to authenticate the Host service in some fashion.

2. Binding Obligations on UMA Participants

This section defines 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.
2.1. Terminology

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.

**Conformance**
Claimed adherence of a running software program or service with the requirements of one or more of the roles ("authorization manager (AM)", "host", and "requester") defined in the UMA core protocol, as defined in its specification. 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 "authorizing user" role as defined in the UMA core protocol, using and configuring software services that variously fill the "AM" and "host" roles in the protocol. 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 Manager (AM)**
A software service that fills the "authorization manager (AM)" role as defined in the UMA core protocol.

**Authorization Manager (AM) Operator**
A Subject responsible for running and operating an AM.

**Host**
A software service that fills the "host" role as defined in the UMA core protocol.

**Host Operator**
A Subject responsible for running and operating a Host.

**Requester**
A software application or service that fills the "requester" role as defined in the UMA core protocol.

**Requester Operator**
A Subject responsible for running and operating a Requester.

**Requesting Party**
A Subject that uses a Requester 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 Requester 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:

- "'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.

- "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 Requester 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:

- "An individual is a person engaged in an online transaction. Individuals are the first priority of the Strategy."
- "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."
- "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.

### T2. This framework uses the following abbreviations.

- **UMA**
  User-Managed Access, the interoperability protocol defined by the UMA core protocol specification and the other specifications it includes by reference.
- **API**
  Application programming interface.
- **AAT**
  Authorization API token, as defined in the UMA core protocol.
- **PAT**
  Protection API token, as defined in the UMA core protocol.
- **RPT**
  Requester permission token, as defined in the UMA core protocol.

Comments: The AAT is a protocol element that is specific to entities in the AM, requester, and requesting party roles defined by the UMA core protocol. The PAT is a protocol element that is specific to entities in the AM, host, and authorizing user roles. The RPT is a protocol element that is specific to entities in the AM, host, authorizing user, requester, and requesting party roles. The RPT protocol element has a definition that can vary, depending on the UMA token profile in use; thus, any obligations that depend on an UMA token profile specify that token profile by name.
2.2. Obligations of the Requesting Party

**R1.** When the Requester successfully receives access from a Host to a protected resource by wielding a valid "UMA bearer token profile" RPT with a currently valid permission for the type of access sought, the Requesting Party gains a) an obligation to the Authorizing Party to adhere to promises it made in order to get access authorization granted, and b) a further obligation to the Authorizing Party not to delegate to third-party Subjects a wider scope of access rights to the same resource than the Requesting Party itself was authorized for.

Comments: At a previous stage, the Requesting Party asked for a relevant permission from the AM and might have had to provide "promissory" claims asserting willingness, for example, to adhere to data usage constraints imposed by the user. This is precisely the end-to-end access authorization agreement that UMA exists to forge. Accepting access to the protected resource binds the Requesting Party to the terms it agreed to, for example, only reading the resource rather than modifying it, or not selling the resource data to someone else if that is what it agreed. The downstream restriction on scope of access is an attempt to put a limited form of "chain-link confidentiality" (and other chain-linked constraints) into effect, as discussed in [ChainLink].

Issues: Note that the obligation goes into effect the first time a Requester gains access under the power of a "currently valid permission". If there was more than one valid permission attached to different sets of promises, without a record (kept by the Host and/or AM) of which permission it used for granting access, ambiguity is introduced. Also note that defining and using token profiles other than the "UMA bearer token profile" might lessen the potential ambiguity.

**R2.** When the AM issues an AAT to a Requester, the Requesting Party using that Requester gains an obligation to the AM Operator to truthfully supply any self-asserted claims required for access authorization at this AM.

Comments: At a later stage, the Requesting Party might be asked to provide claims to support authorization processes at this AM, for accessing all resources protected by this AM, managed by any Authorizing Parties, at any Hosts. The Requesting Party's responsibility to provide truthful claims in all these cases begins now. The Requesting Party can remove this obligation by revoking the AAT.

**R3.** When the Requesting Party provides claims to an AM to satisfy the AM's authorization process, the Requesting Party gains an obligation to the AM Operator to truthfully supply any claims required for access authorization for the requested permission.

Comments: This clause is specific to the action of providing actual claims, as opposed to clause R2. Where the Requesting Party supplies claims in a form that can be verified by the AM, 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.

Issues: Is this clause really about an obligation to the AM Operator, or ultimately to the Authorizing Party? Claims gathered at this moment could, for example, be promissory: "I agree not to release this data publicly until after embargo date XYZ." Would this just be a repeat of clause R1? Do we even need R3 given that we have R1 and R2 in the mix?

**R4.** When the AM issues an RPT to a Requester, the Requesting Party using that Requester gains an obligation to the Host Operator to represent the legitimate bearer of the RPT whenever it presents this token to the Host.

Comments: In the case where the "UMA bearer token profile" is being used, the token cannot be bound in any meaningful way to the specific requester and requesting party it applies to, so the Requesting Party takes on the obligation of protecting the RPT from theft and not maliciously sharing the RPT to be used by others. Defining and using different UMA token
2.3. Obligations of the Host Operator

**H1.** When the AM issues a PAT to a Host, the Host Operator gains an obligation to the Authorizing Party to delegate protection services to the AM Operator for the set of protectable resources for which it represents this capability to the Authorizing Party, and to respect the access constraints represented by RPTs generated by the AM.

Comments: Once the AM Operator becomes the Authorizing Party’s authorization proxy, it begins relying on the Host Operator in other, more specific ways. The Host has the opportunity to inspect AM-issued permissions or take other actions that delegate protection responsibility to the AM at a later stage, but its responsibility for respecting them begins now. The specific protection services made available to the Host by the AM differ depending on the "UMA token profile" in use. The Authorizing Party can remove this obligation by revoking the PAT.

**H2.** When the AM issues a PAT to the Host, the Host Operator gains an obligation to the AM Operator to register resource sets and applicable actions accurately and timely according to the Authorizing Party’s expressed instructions for protection.

Comments: At a later stage, the Host has the opportunity to register resource sets, but its responsibility for performing this task begins now. The Authorizing Party can remove this duty by revoking the PAT. The Host Operator may have contracted with the Authorizing Party for service-level agreements to respond specifically to timeliness needs and so on.

**H3.** When the Host checks the status of an "UMA bearer token profile" RPT, the Host Operator gains an obligation to the AM Operator to respect the status of permissions granted by the AM in allowing or disallowing access by the Requester.

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

**H4.** When the Host responds in any fashion to a Requester’s access request, the Host 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 Host Operator, as a Subject that is otherwise potentially not obligated to the AM Operator, carries a great deal of responsibility here to make every effort to grant access where the AM has granted permission. Its interpretation of scopes and permissions, particularly in the case where the RPT presented by the Requester uses the "UMA bearer token profile", is not entirely auditable by the requester or AM. Further, issues can arise from the skew between permission validity periods and actual access. Defining and using different UMA token profiles can mitigate the risk of failure on the Host Operator’s part.

2.4. Obligations of the AM Operator

**AM1.** When the AM issues a PAT to the Host, the AM Operator gains an obligation to the Authorizing Party to represent the Authorizing Party's policies accurately and timely in
granting permissions.

Comments: At a later stage, the AM will require the Host to present the PAT whenever it uses the AM’s protection API on behalf of this Authorizing Party. The Authorizing Party can remove this obligation by revoking the PAT. The AM Operator may have contracted with the Authorizing Party for service-level agreements to respond specifically to timeliness needs and so on.

**AM2.** When the Host registers a permission at the AM, the AM Operator gains an obligation to the Host Operator to represent the Authorizing Party’s authorization policies accurately and timely in granting permissions.

Comments: At a later stage, when a Requester approaches the AM seeking a previously registered permission, the AM matches Authorizing Party policies to the requested permission to drive any requests for claims and ultimate access authorizations, but its responsibility for performing this task begins now.

**AM3.** When the AM issues an AAT to a Requester, the AM Operator gains an obligation to the Requesting Party to request only claims that support the purpose of satisfying the user’s policy.

Comments: At a later stage, the AM might ask the Requesting Party to provide claims for specific permission purposes, but its responsibility begins now. The Requesting Party can remove this duty by revoking the AAT. When it is possible in OAuth to perform token revocation, the Requester Operator can remove this duty itself by revoking the AAT.

---

**2.5. Obligations of the Authorizing Party**

**AP1.** When the AM responds positively to a Requester’s request for a permission, the Authorizing Party gains an obligation to the Requesting Party using that Requester to respect the boundaries of data usage constraints placed on the Requesting Party in the form of requests for claims driven by the Authorizing Party’s policy at the AM.

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 the granted permission.

**AP2.** When the AM issues a PAT to a Host, the Authorizing Party gains an obligation to the AM Operator to introduce the desired Host to this AM in outsourcing protection of this host’s resources.

Comments: How the Host learned of the AM’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 AM before the AM successfully issues the PAT. The Authorizing Party can remove this obligation by revoking the PAT.

**AP3.** When the AM issues a PAT to a Host, the Authorizing Party gains an obligation to the Host Operator to introduce the desired AM to this Host in outsourcing protection of this host’s resources.

Comments: Once the AM Operator becomes the Authorizing Party’s authorization proxy, the Host Operator begins relying on it in other, more specific ways. How the Authorizing Party indicated the desired AM 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 AM before the AM successfully issues the PAT. The Authorizing Party can remove this obligation by revoking the PAT.
3. Acknowledgments

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

- Domenico Catalano, Oracle Corp.
- Kevin Cox, Edentiti
- Sal D'Agostino, IDmachines
- Susan Morrow, Avoco Secure

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. Change History

The following changes were made from 00a to 00b:

- Terminology went from Entity to Individual, Non-Person Entity (NPE), and Subject. NSTIC document referenced as part of this change.
- Doc now consistently refers to itself as a "document" (not a specification) with a "contractual framework" embedded within it.
- Removed "constellation" language as not adding value.
- Removed the single Requester Operator clause since it's already covered in the out-of-scope obligations section.
- Added clause-specific issues.
- Removed stray mentions of "duties" (vs. "obligations").
- Removed mentions of autonomous token revocation. When OAuth standardizes this, we can account for it.
- Clauses generally cleaned up and made more consistent.

The following changes were made from 00b to 00c:

- Cleaned up some items identified by Dazza.
- Ensconced mentions of the "UMA bearer token profile" more cleanly, and removed clause-specific issues having to do with token profile specificity.

5. References

5.1. Normative References


5.2. Informative References
Appendix A. Document History

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

Author's Address

Eve Maler (editor)
XMLgrrl.com

Email: eve@xmlgrrl.com