Legal Considerations in UMA Authorization

> *"Privacy is not about secrecy. It's about context, control, choice, and respect." (UMA webinar, after Bob Blakley)*

"

Abstract

This document explores legal issues raised by the act of using User-Managed Access (UMA) to authorize another party to get web resource access.

Status

This document is a product of the User-Managed Access Work Group. It is currently under active development. Its latest version can always be found here. See the Change History at the end of this document for its revision number.

Editors

- Eve Maler
- Tom Smedinghoff ??
- Mark Lizar

TBD

Add sharing scenario.

Table of Contents

Introduction

User-Managed Access (UMA) allows a user to make access demands as well as manage access to their own resources outside of each and every service provider . The implications of these demands quickly go beyond cryptography and web protocols and into the realm of rights, contracts and liability.

UMA is a protocol for access control to resources at a host. E.g. a Internet Service provider.

Such a service should allow a user to control data-sharing and service-access relationships between online services hosting and accessing data. An external User controlled access manager requires the ability to reside in distinct domains and establish relationships between services in a dynamic way. For the access relationship service to be usable across multiple web applications, it should not be required to understand the representations of resources it is charged with protecting and its functionality should be applicable to arbitrary web resources.

UMA is developed to give a web User a unified control point for authorizing *who* and *what* can get access to his or her online personal data (such as identity attributes), content (such as photos), and web-based services (such as viewing and creating Twitter-style status updates), no matter where all those things "live" on the web. Further, UMA allows the user to make demands of the other parties to the transaction with which these parties must comply in order to for their request for authorization to view/consume the user's data to be approved. These demands can include requests for information (such as "Who are you?") and promises (such as "Do you agree to these Non-Disclosure Agreement terms?").

UMA targets end-user convenience and development simplicity as goals. But it also seeks *enforceability* of authorization agreements, in order to make the act of granting data and service access truly informed, un coerced, and meaningful – no longer a matter of mere passive consent but rather a step that more fully empowers ordinary web users.

For all these reasons legal considerations need to be aware of the impact UMA has on issues related to authorization policy, contracts, liability, and enforceability that arise among the various actors in UMA interactions.

(Note: This document is in the process of being edited to be accessible to readers, even relatively nontechnical ones, who have expertise in these areas, and we welcome suggestions for improvement.)

Starter Scenarios

Let's imagine a web user, **Alice Adams**, who doesn't mind sharing her personal travel information with the right sources as long as the process of sharing

- Is convenient (e.g., no requirements to alert authorized recipients every time a new trip gets booked);
- meets her expectations for the uses of that information are met (e.g., she's not worried about recipients divulging to the whole world that her house is going to be empty);
- Provides control mechanism dictating what's allowed and not allowed (e.g., she can get an at-a-glance view of who's seeing what).

She uses a website called **TravelIt.com** (think TripIt) to store all of her travel itineraries. Since it's the nature of travel information to change frequently, she wants to be sure that her good friend **Bob Baker** always has the latest version so he can pick her up from the airport on time and make sure her cat is fed while she's away; he likes to use **Schedewl.com** (think Google Calendar) for subscribing to TravelIt. She would also like her social travel site **Airplanr.com** (think Dopplr) to pick up her itineraries automatically and make them available to friends who are on that system.

Because Alice is a frequent and seasoned traveler, she's interested in entertaining discount offers from travelogue company **FrodoReviews.com** (think Frommer's) for making her itineraries available to them for survey purposes.

To this picture, UMA adds the possibility of a new kind of web-based application: a kind of "traffic cop" for overseeing all these instances of travel itinerary sharing, which will help Alice manage her digital footprint. We'll call this site **CopMonkey.com**.

(am thinking of re-vamping this scenario so that it refelects an external access control manager than that of the Host (service provider) Mark ??

UMA overview and terminology

As hinted in the introduction, UMA involves four main players:

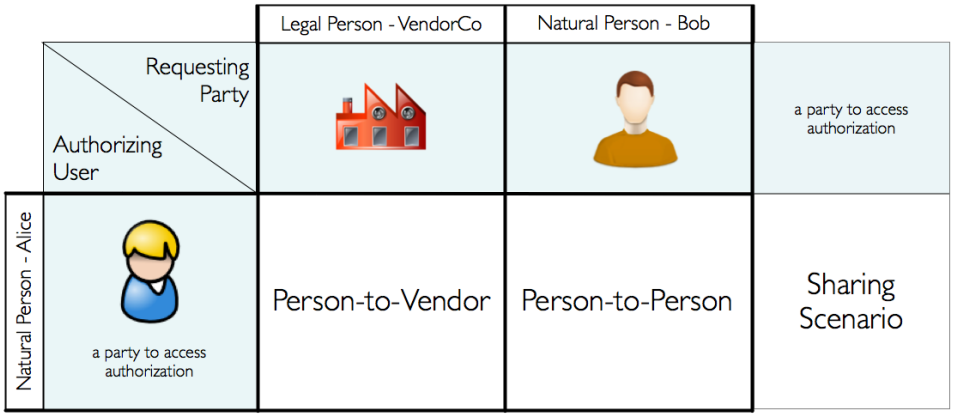
- Alice is an **authorizing user**: a web user who gets to be in charge of authorizing access to his or her "stuff" on the web (known as **protected resources**).
- The TravelIt application acts as a **host**: one of possibly many websites where Alice's resources are stored and managed in various fashions.
- The Schedewl, Airplanr, and FrodoReviews applications act as **requesters**: web-based applications that seek access to the authorizing user's protected resources. Access may mean more than just downloading or viewing, for example adding more "stuff" to be stored, or manipulating or transforming it in some other way.
- The CopMonkey application acts as an **authorization manager** or **AM**: a unified web-based control point that makes it easy for Alice to set up protections for all those resources at their various hosts, and to control and track access to them by requester's.

- *# Establish general policies for the access and protection of all her protected resources at their various hosts,
    1. Set up unique individual rules for specific combination's of protected resources, hosts and requester's,
    2. track access to her protected resources by requester's and
    3. manually manage specific interactions as established in her policies.

So far, this description relates only to the UMA web protocol itself. These four players are commonly referred to as "endpoints" in discussions of computing protocols (or sometimes as "entities" or "parties", but we'll avoid these in order to avoid confusion with the legal connotations of these words).

You might be asking, Where did Bob go? Recall that the authorizing user can make demands of the counter parties (Bob is an other party like Schedel. com, and Airplanr.com) to judge their suitability for access. We must ask: What is the nature of the "counter parties " like Bob? The authorizing user is flesh-and-blood, but the AM, host, and requester endpoints are just *tools*. They are implemented in software, and the software is deployed in the form of networked applications and services. A tool can't be responsible for the consequences of accessing an authorizing user's "stuff". For this reason, we introduce another important player that's not strictly part of the UMA protocol:

- In our examples, Bob and the companies that run Airplanr and FrodoReviews are **requesting parties**: legal persons (such as corporations) and real human beings (other than Alice herself) who use a requester endpoint to seek authorized access to some protected resource.



We'll explore this complex set of players more in a moment.

The policy capabilities of UMA allow the authorizing user to configure the AM with policies that constrain the conditions for access by others in advance, so that the authorizing user need not be present (online) when transactions with requesters take place. Then, when a requester endpoint comes calling, the policy may compel the AM to ask the requestor to convey the following:

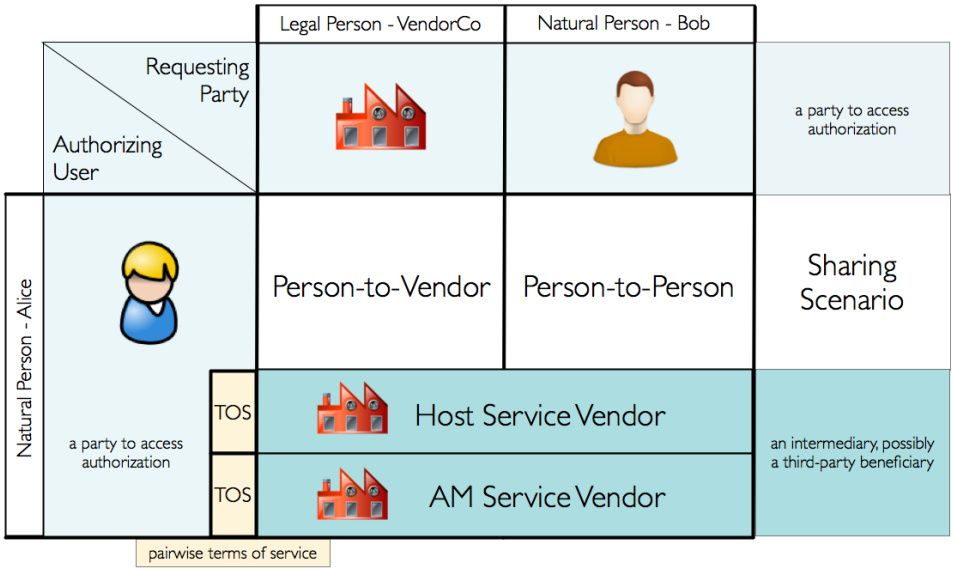
- A set of **claims**: i.e. one or more affirmative or promissory statements about the requesting party that are intended to satisfy some policy Alice has configured into her AM.

Now we have enough language to begin discussing potential access authorization *agreements* and *liability* that may obtain between two parties (yes, in the legal sense) interacting in an UMA environment: the authorizing user and the requesting party.

NEED TO EDIT FROM HERE ****

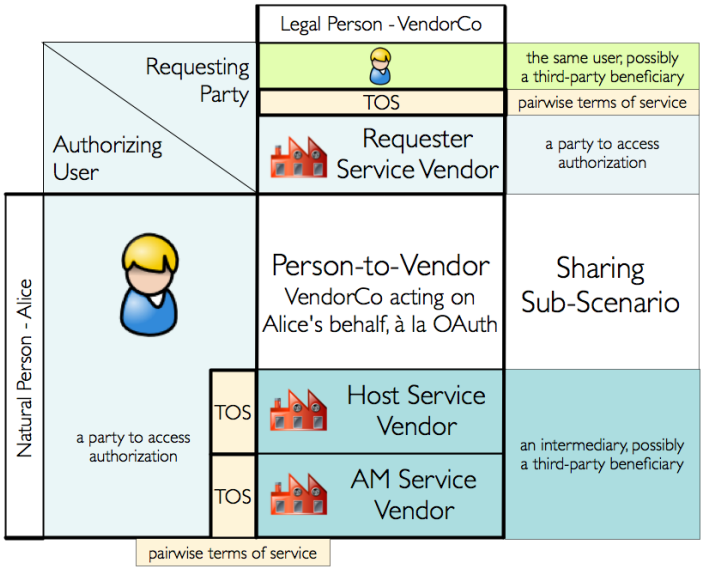
Intermediaries the authorizing user may employ in protecting resources

Alice likely is an ordinary user of the web, and tends to use web applications written by third parties rather than hacking her own services and deploying them on the networked workstation that sits under her home-office desk. Thus, she chose to use the third-party services run by the TravelIt and CopMonkey companies, and in creating accounts with them, she likely had to agree to (or negotiate) **terms of service** with each of these companies. They are thus **intermediaries** in providing UMA protection to her resources, rather than parties to actual authorized access.



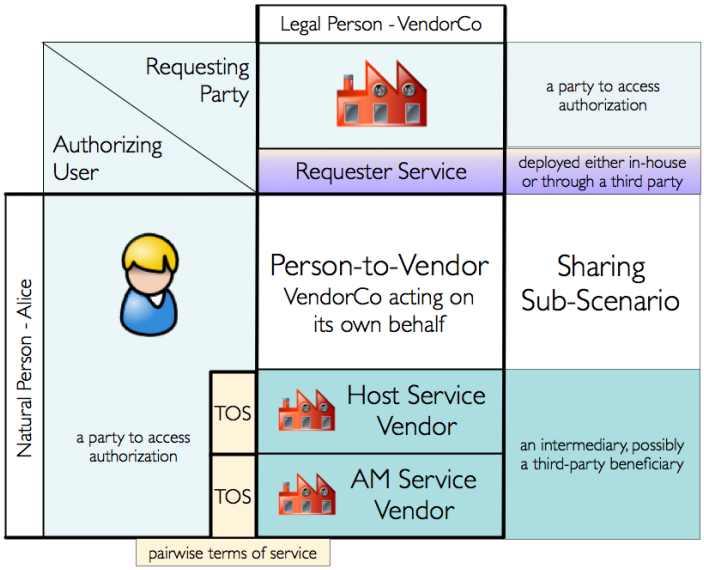
Sharing with Airplanr

@@TBS



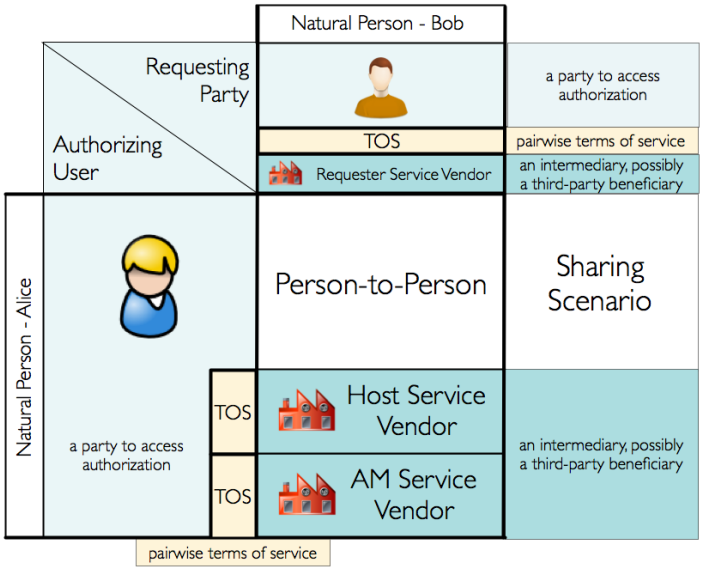
Sharing with FrodoReviews

@@TBS



Sharing with Bob through Schedewl

@@TBS



The nature of claims

A claim may be *affirmative*, representing a statement of fact (as asserted by the requesting or another claim issuer); or *promissory*, a promise (as asserted by the requesting party specifically to the authorizing user). A statement of fact might be "The requesting party is over 18 years of age." A promise might be "The requesting party will adhere to the specific Creative Commons licensing terms indicated by the AM." There are technical dimensions to expressing and conveying claims, but since UMA strives to provide enforceability of resource-access agreements, there may also be legal dimensions.

In cases where a claim constitutes acceptance of an access-sharing contract offer made by the authorizing user (as presented by the AM as his or her agent in requiring the claim), the authorizing user and requesting party are the parties to the contract, and all other legal or natural persons running UMA-related services involved in managing such access are intermediaries that are not party to the contract (though they might end up being *third-party beneficiaries* in some cases).

Where the primary resource user and the authoring user differ, there is likely to be an interaction (invisible to UMA) at the host service that allows (or forces) the primary resource user to designate an authorizing user, and an agreement that the authorizing user acts as the primary resource user's agent or guardian or similar.

Mapping requesting parties to profiles for getting access tokens

We suspect that certain sharing patterns lend themselves to choosing different profiles for UMA's step 2 (getting a token).

- When Alice authorizes sharing of her TravelIt resource with Airplanr (in the context of "herself again" logged into Airplanr), a user authorization flow might possibly be appropriate as part of the process of Airplanr satisfying CopMonkey that it's really Alice again. ??
- When Alice authorizes sharing of her TravelIt resource with FrodoReviews acting on its own behalf, an autonomous client flow seems appropriate.
- When Alice authorizes sharing of her TravelIt resource with Bob through Schedewl, we're not sure if this should use an autonomous client flow or if something similar to it needs to be written specially to add the "on behalf of a totally different person" semantic.

Change History

Version	Date	Comment
Current Version (v. 11)	**Oct 28, 2010 12:51**	**Mark Lizar**: Migration of unmigrated content due to installation of a new plugin
v. 10	Oct 28, 2010 12:51	**Mark Lizar**: Migrated to Confluence 4.0
v. 9	Oct 28, 2010 12:51	**Mark Lizar**
v. 8	Oct 14, 2010 13:50	**Mark Lizar**
v. 7	Sep 30, 2010 13:34	**Mark Lizar**
v. 6	Apr 14, 2010 01:30	**Eve Maler**
v. 5	Apr 14, 2010 01:26	**Eve Maler**
v. 4	Apr 14, 2010 01:15	**Eve Maler**
v. 3	Apr 11, 2010 18:47	**Eve Maler**
v. 2	Apr 11, 2010 16:18	**Eve Maler**
v. 1	Apr 08, 2010 19:12	**Eve Maler**