UMA telecon 2009-10-22

UMA telecon 2009-10-22

- Date and Time
- Attendees
- Regrets
- Agenda
- Minutes
  - Roll call
  - Approve minutes of UMA telecon 2009-10-15
  - Action item review
  - Spec progress and review (Paul)
  - Scenario review (docket)
- Next Meeting: UMA telecon 2009-10-29

Date and Time

NOTE: This month has a lot of summer-winter time changes in it. Please take careful note of your local hour.

- Day: Thursday, 22 Oct 2009
- Time: 9:00-10:30am PDT | 12:00-1:30pm EDT | 16:00-17:30 UTC (time chart)
- Dial-In:
  - Skype: +9900827042954214
  - US: +1-201-793-9022 | Room Code: 295-4214 (other local country numbers available on request)

Attendees

Voting participants (quorum is 11 of 20):

1. Adams, Trent
2. Akram, Hasan
3. Bryan, Paul
4. Davis, Peter
5. Fletcher, George
6. Hanson, Michael
7. Hogberg, Jonas
8. Maler, Eve
9. Scholz, Christian
10. Stollman, Jeff

Guest:

- Joni Brennan (staff)

Regrets

- Joe Andrieu
- Iain Henderson
- Mark Lizar

Agenda

- Roll call
- Approve minutes of UMA telecon 2009-10-15
- Action item review
- Spec progress and review (Paul)
  - Proposal for "Step 0" host registration at AM sent in email
- Scenario review (docket)
  - Formally accept/reject aspects of the requester delegate scenario
  - Discuss distributed services scenario and its mass-authorization aspects
- 2 Nov 2009 F2F planning
- AOB

Minutes

Roll call

Quorum was not reached.

Approve minutes of UMA telecon 2009-10-15
Deferred due to lack of quorum.

**Action item review**

- 2009-10-15-2 Eve Open Write a "Hey, Sailor" scenario to illuminate the needs around Requesters that ask for resource access without the User expecting them. (Still open.)
- 2009-10-15-3 Gerry Open Write an hData scenario for the Scenario document. (Still open.)

**Spec progress and review (Paul)**

- Proposal for "Step 0" host registration at AM sent in email

Paul would like to establish some agreement today (informal because we're not quorate) that XRD and LRDD are the right technologies to utilize for dynamic host registration.

LRDD ("lard") is a way of discovering the location of a descriptor about a resource. If the resource is an HTML document, it can embed a link with a special rel. (OpenID uses this trick.) Or if the server delivering the resource knows where the metadata is, it can embed a header in the HTTP response. There's also a "known location" approach you can fall back onto, and a "known pattern" to discover the metadata locations of other resources. And there's a way to use email addresses. The webfinger folks have been talking about a way to push this into the DNS record. LRDD is generic and doesn't depend on XRD as as the metadata to be discovered (SAML metadata, (G)RDDL, POWDER), but Eran Hammer has been working on it with the intent of using a rel type of XRD, partly so that OAuth can use all this for dynamic OAuth service discovery.

(Paul had been concerned, looking at the full (original) XRDS spec, that it was trying to replicate DNS too much, and this was uncomfortable from a RESTafarian perspective.)

Partly for flexibility, extensibility, and dynamism, and partly because it seems to be the right way to align with other related efforts, Paul supports using XRD and LRDD. Michael expressed agreement.

Walking through Paul's proposed text:

In the examples, "oauth.net" values refer to real OAuth endpoints, and (fictitious) "uma-wg.net" values refer to proposed UMA endpoints.

n. Host registration [excerpt]:

The user should be able to just type in (or select an icon for, etc.) "copmonkey.com" at a host, and the correct OAuth and UMA metadata about it needs to be progressively discovered/disclosed to this host.

n.1 User introduction:

Here the Rel of "/host" is a type of endpoint, and its corresponding URI is the endpoint where this host should go next.

n.2 Access token issuance:

The host tries to register itself with the AM, by "creating itself" at the AM using POST. It gets challenged, and the challenge comes with metadata. At this point we're into OAuth proper, and this involves user confirmation etc. in all the same ways at OAuth.

n.3 Request host resource creation (w. signature):

Once the AM verifies the signature successfully, it can create the unique resource to be used by that host. Before Paul started using the XRD/LRDD paradigm, he put some direct metadata transfer into this step. But that led to defining new syntax and such, and didn't allow for flexibility to periodically change and add to the endpoints and the "host handle" it makes available to that host.

n.4 Retrieve host resource metadata:

This step shows the full force of what can be conveyed by the XRD, and in what manner. Because this XRD ultimately expires and needs to be refreshed, it's possible to move resources around, add new version support, and so on. The host-specific resource created at the host's request is static, but the full interface definition for that resource is dynamic by virtue of XRD expiry.

One thing Paul is still working on is a new field in the XRD that will hold the AM URL that the host should tell requesters to go to in order to register with the AM. Michael describes this as a way for the AM to make the host handle discoverable.

George questions why we need XRD for this, vs. just having a standard API at the AM. Paul didn't want to have to invent a new schema for the data, and found it useful to lean on the ability to refresh the values set in the XRD. In fact, as noted in his original email, Paul is even tempted to overuse the handy ability of XRDs to convey pairs of generic names/types and values! George finds the dynamism attractive.

Are there any privacy issues around this proposal? Yes. The way the AM should behave (vs. must?) is that it only lets hosts who created an AM resource to access it. However, the host handle doesn't seem secret. And even the request for the XRD needs to be signed (we believe it can happen over plain HTTP).

Actually, any authentication mechanism between the host and the AM would work -- should we not mandate OAuth vs. something like HTTP Basic Auth, just strongly recommend it? We consider OAuth the only game in town for the moment, but we want to future-proof it in case really good successors arise. ("Past-proofing" it by suggesting that HTTP Basic would be acceptable is not our goal!) The natural latency between revisions of specs that reference each other suggests that we should add flexibility here; e.g., look at how XML V1.0 needed a Second Edition to handle the publication of Unicode V3.0. Paul suggests that we provide a spec appendix that shows an end-to-end working example, showing OAuth in its rightful role as our preferred mechanism.

How would the AM discover facts about each host? This still needs to be added. This may come into play for:
What are the URLs for the resources being protected at this host?
What content types does the host handle?
What filtering/content-type-specific policies does the host offer?
How to find auxiliary protected-resource metadata (such as thumbnails for photos)?
Etc.

At one time, there was discussion about "simple OAuth", which had the notion of a token issuer. George comments that he hasn't seen any updates on this for a while now; the private conversations referred to in a previous meeting are related to this. The token issuer in that conception was a little similar to our externalized AM function, but it interfered with some security provisions in OAuth and would issue very short-lived tokens in order to mitigate this. One purpose of the proposal was to reduce the complexity of the OAuth protocol by dropping signing, and this is where George's security concerns lie. And the other purpose was to describe an extension to OAuth that lets the consumer/requester just authenticate to the token issuer. They seem to be a little stuck on how to make it cross-vendor at this point.

(This seems similar to Christian's distributed-services scenario, below.)

Peter notes, with his XRI TC chair hat on, that the XRD spec is going into last call, and any comments should be provided ASAP.

Summary: Paul recommends XRD and LRDD, and it affects the design process going forward if we don't want to use it. The people on the call generally express support for the direction. Also, we agree that OAuth is the "incumbent" way for hosts to authenticate to AMs but we want to future-proof the spec in this respect.

- Proposal for introduction/terminology section sent in email

Deferred due to lack of time.

Scenario review (docket)

- Formally accept/reject aspects of the requester delegate scenario

Deferred due to lack of quorum.

- Discuss distributed services scenario and its mass-authorization aspects

Eve's biggest question is whether the "New Service" referred to in the scenario is approaching the AM as a host (asking to be registered in the catalog) or is approaching as a requester (asking to get pre-authorized to all the services to do whatever it is they allow)? It could be both, but the primary purpose is the latter.

Christian describes a case of the latter as: You want to join a new social networking site, but don't want to have to load it up manually with your account registration data, your existing social graph, your existing contact book, etc. This is essentially the "data portability" scenario! (Keep in mind that the service catalog aspects of the scenario are incidental to the mass authorization use case. We definitely don't want to solve for service catalogs but rather reuse /assume someone else's solution, like the webfinger one.)

Do we need an offline authorization request mechanism? This could be leveraged for solving mass authorization.

The inefficiency in just using regular OAuth for authorizing a requester's desire to connect to (say) ten hosts is that the user has to get redirected ten times in order to set up the connection. But mass authorization would, of necessity, have to grant more coarse-grained access to the set of hosts. But where the API of the host is well-known (like with PoCo), maybe it's possible for an AM to give the user finer-grained control of access settings as a value-add feature (or using some UMA features we haven't invented yet?).

Paul thinks we can maybe solve this simply by writing the terms for a resource in a particular way. Maybe you can state a policy that says "Anybody who has a Google certificate has a certain kind of access to a certain protected resource." This is pre-authorization by policy! You can also have a default policy and set of terms, sort of the "anonymous" case. We still need to do a bit more thinking about whether we need "hardcore" pre-authorization, though. Eve notes that ID-WSF allows its Discovery Service to do the latter, and she believes some r-card approaches allow this too.

Christian suggests that policy stuff and the terms mechanism should be in a separate spec.

Als:

<table>
<thead>
<tr>
<th>Eve</th>
<th>Open</th>
<th>Follow up with George on understanding the ongoing token-issuer discussions and contributing to them in the IRW timeframe.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paul</td>
<td>Open</td>
<td>Provide a breadth-first fledgling &quot;example appendix&quot; for 2009-10-29.</td>
</tr>
<tr>
<td>Christian</td>
<td>Open</td>
<td>Write up a proposal as discussed on his previous call with Paul and Eve.</td>
</tr>
</tbody>
</table>

2 Nov 2009 F2F planning

- We should decide on the mass-authorization issue.
- We should decide the disposition of other major scenarios with distinctive aspects.
- During that week, we should reach out to others who seem to be working on similar things.

Next Meeting: UMA telecon 2009-10-29

NOTE: October 29th is definitely "in between" North American and European seasonal time changes! For most of Europe, the meeting time will temporarily move to 5pm!

- Day: Thursday, 29 Oct 2009
- Time: 9:00-10:30am PDT | 12:00-1:30pm EDT | 16:00-17:30 UTC (time chart)
• Dial-In:
  • Skype: +9900827042954214
  • US: +1-201-793-9022 | Room Code: 295-4214 (other local country numbers available on request)