UMA telecon 2010-04-15

Date and Time

- **Day:** Thursday, 15 Apr 2010
- **Time:** 9:00am-10:30am PST | 12:00-1:30pm EST | 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)

Agenda

- **Roll call**
- **Approve minutes of 2010-04-08 meeting**
  - Upcoming meeting schedule: telecons on Apr 22, 29; no telecon on May 6 due to public workshop on May 4...
- **Action item review**
- **Implementation focus team report**
- **Meta-discussion of scenario approval process**
  - A smaller team is meeting Wednesdays for the next few weeks to get ready for mid-May activities
- **Protocol issues review and OAuth 2.0 input**
  - Token validation
  - Scope
- **AOB**

Attendees

As of 7 Apr 2010, quorum is 6 of 11.

Adams, Trent
Akram, Hasan
Armstrong, Brian

1. Bryan, Paul
2. Catalano, Domenico
3. Fletcher, George
4. Holodnik, Tom
5. Machulak, Maciej
6. McIntosh, Michael
7. Maler, Eve
8. Moren, Lukasz

Non-voting participants:

- Thomas Hardjono
- Mark Lizar
- Nat Sakimura

Regrets:

- Brian Armstrong

Minutes

**New AI summary**

<table>
<thead>
<tr>
<th>2010-04-15-1</th>
<th>Eve</th>
<th>Open</th>
<th>Reach out to legal eagles to get review of new technical report.</th>
</tr>
</thead>
</table>
Roll call
Quorum was reached.

Approve minutes of 2010-04-08 meeting
Minutes of 2010-04-08 meeting APPROVED.

- Upcoming meeting schedule: telecons on Apr 22, 29; **no telecon on May 6** due to public workshop on May 4.
- A smaller team is meeting Wednesdays for the next few weeks to get ready for mid-May activities; others are welcome; just send Eve email to get the dial-in information.

Action item review

- **2010-03-10-2 Maciej Open** Do next round of spec editing. Eve has edited the terminology section to put it on an OAuth 2.0 basis, and Maciej hopes that we’ll make progress today towards deciding matters that will allow him to edit substantive parts of the spec.
- **2010-03-25-1 Paul/TomH Open** Send email giving examples of how a resource-oriented scope approach is necessary. Tom will send out his writeup to the whole list.
- **2010-04-01-3 Eve Closed** Create a new technical report draft on legal considerations in UMA authorization scenarios. There is a draft in place. It needs work.
- **2010-04-08-1 Eve Open** Turn Paul’s “claims 2.0” proposal into a draft spec. Still pending.
- **2010-04-08-2 TomH Open** Revise the tax scenario for inclusion in the Scenarios document. Still pending.

Dynamic introduction, discovery, and trust

As discussed by the focus team yesterday, host->AM introduction is incomplete: In the current spec, the authorizing user can introduce the host to an AM, but according to the WRAP/OAuth paradigm (with only static introduction), the host needs to be provisioned ahead of time with a client ID and secret. We have not accounted for this. In order to allow for dynamic introduction, the host needs to get provisioned with these two pieces of information. Specifically, it has to get them before it has to enter into the user authorization flow, but around the time of – presumably just after – it finds the AM's hostmeta.

Last week we discussed the relative priorities in our various interests in UMA. TomH is more interested in the claims exchange than in dynamic introduction, but Eve sees the latter as an important value-add on top of core OAuth functionality, even if the high-trust scenarios still want static introductions, whitelisting, etc.

It sounds like no one is arguing for UMA 1.0 to **not** have this feature. But can we solve it for the May timeframe? Originally, the spec used the technique of having the AM publish an “anonymous” key and secret that every host could use, but Eve didn’t (oops) carry over this solution – or any solution – into the WRAP-based version of the spec.

If we use the anonymous-credentials approach, they could simply be openly published in the hostmeta, with no request-response pattern needed to have the host specially get a unique client ID and secret to use with the AM. Since the access token it gets eventually (assuming the authorizing user has actually authorized this connection) is unique to that host anyway, the client credentials don’t have to be unique. Or are there extra security considerations around this, since the client credentials would need to be used (in addition to the access token) whenever the host needs to communicate further with the AM?

George points out that an alternative approach could be for the AM to host the send it some sort of host-created client credentials, say, based on its realm (so that it doesn’t clash with credentials created by other hosts).

It seems that the technical proposition of solving dynamic discovery, apart from trust issues, is simply not that hard. The two ends of the continuum are:

- No need for special AM/host trust; published anonymous credentials; totally dynamic discovery
- High need for AM/host trust; totally static discovery through prior assignment of pairwise AM-host client credentials

Beyond these two choices, perhaps we should learn through experience whether any nuanced middle-ground solutions are needed, such as publishing a credentials negotiation endpoint in the hostmeta and inventing a protocol for using it. George wonders if we can even reuse something like the OAuth assertion flow for this purpose, since it requires just a SAML assertion vs. OAuth-native client credentials. Since this particular flow doesn’t involve a user directly, maybe we’d have to define a variant that does include the user authorization piece.

Eve points out that the hData scenario needs dynamic introduction of physicians’ office apps to the AM (and discovery service), because the user gets to choose their own physician, but the AM/Disco will want to demand claims from that host. So in that case, step 1 is like an embedded UMA flow vs. a plain embedded OAuth flow! Maybe that is the way to go.

We have unanimous **CONSENSUS** that we’ll add anonymous credentials to the hostmeta for the “no need for trust” case, and we’ll rely on out-of-band mechanisms for the “high need for trust” case for now.

**Singular user authorization URL in current OAuth 2.0 draft**

As discussed in yesterday’s focus team meeting, there is now just a single endpoint for all client->AS communications (for us, requester->AM). UMA has a design goal to be RESTful and resource-oriented, but it seems OAuth is going in the direction of being less resource-oriented. Are we okay with building directly on top of an OAuth that may not be perfectly elegantly defined according to RESTful lights?

Eve would like to influence the OAuth design process as much as we can, based on our own ideas of good design, but sees the value of building directly on top of OAuth as a “black box” as so huge that it trumps perfect RESTfulness of our substrate. There seems to be general agreement.
In fact, last week Eve had been advocating that we move onto an OAuth 2.0 basis as quickly as possible, even though it means certain details will keep changing out from under us, so that we can do our best to influence the OAuth 2.0 design process.

We have unanimous CONSENSUS that we’ll use the April 12 draft of OAuth 2.0 as our basis for the IIW timeframe, unless we explicitly use an exception process to change this.

Meta-discussion of scenario approval process

We’re agreed that we will return to serious scenario disposition work in the late-May (post-IIW) timeframe, so we can learn the boundaries of our UMA 1.0 scope.

We’ll share user stories and wireframes on the WG list so that we can coordinate these activities in email.

Protocol issues review and OAuth 2.0 input

The OAuth list has been heavily discussing token size (which we believe relates to token validation methods) and scope (including how to upgrade scope). Currently they have removed the scope parameter in favor of an HTTP realm approach.

Eve did float the current UMA solution on the OAuth list, and its RESTful approach seemed to resonate with at least a couple of people there. Are we satisfied that, at a minimum, OAuth allows us to cleanly extend it to add scope stuff we want? We’re not sure. What if we have to invent our own scope parameter and include it in some HTTP header? Will that work okay? TomS is not sure.

We do think that the scope upgrade feature (“scope creep”? 😁) is something for which there are classic OAuth use cases as well as many enterprise-related use cases.

Next Meeting: UMA telecon 2010-04-22

- **Day:** Thursday, 22 Apr 2010
- **Time:** 9:00am-10:30am PST | 12:00-1:30pm EST | 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)