UMA telecon 2010-09-09

Date and Time
- WG telecon on Thursday, 2 Sep 2010, at 9-10:30am PT (time chart)
  - Skype line "C": +9900827042954214
  - US: +1-201-793-9022 | Room Code: 295-4214

Agenda
- Administrative
  - Roll call
  - Approve minutes of 2010-08-26 and 2010-09-02 meetings
  - Action item review
- Scenarios and use cases
  - Review Mario's "distinctive aspects" classifications
  - Select "canonical scenario" for next phase of design work
- Resource/scope registration
  - Review Domenico's UX explorations
- Step 3
  - Discuss Christian's proposal
- AOB

Attendees
As of 02 Sept 2010, quorum is 6 of 10.
1. Hardjono, Thomas
2. Holodnik, Tom
3. Maler, Eve
4. Scholz, Christian

Non-voting participants:
- Kevin Cox
- Alam
- Jeff Stollman
- Herve Ganem
- Mark Lizar

Regrets:
- Maciej Machulak
- Lukasz Moren
- Mario Hoffmann
- George Fletcher

Minutes

Roll call
Quorum was not reached.

Approve minutes of 2010-08-26 and 2010-09-02 meetings
Deferred due to lack of quorum.

Action item review
Scenarios and use cases

- Review Mario's "distinctive aspects" classifications
- Select "canonical scenario" for next phase of design work

Deferred due to Mario's absence. Eve notes that choosing a scenario with a "person-to-self" sharing aspect would be simple enough and yet realistic enough to spec out something useful. It's for this reason that she's been suggesting the location services scenario. Tom and others speak in favor. Location services are also interesting because they often involve requesting read/write access, which has scope wrinkles. Tom suggests that this scenario has good privacy implications because it allows a person to see globally where the information is spreading to.

(Eve referenced some blog entries (1, 2) that touch on the subject of requesting access in order to put data into a host, rather than pull data out of it.)

Resource/scope registration

- Review Domenico's UX explorations

**UMA Scope Conceptual Model:** The conceptual diagram shows that policies:scopes are many:many. It's still controversial that resources and scopes are separate here, since the current scope proposal has just a flat list of scope strings that doesn't distinguish between resources and actions on them. Herve observes that if we really have two conceptual dimensions but must squish them into a scalar scope list, we get an explosion of that list.

**Resource Scope as RESTful Web Service:** Is this truly RESTful? We're thinking that a more RESTful approach might be allcalendar.com/usercalendar /aadams/travel, and then GET and POST would have different effects on it. But even so, we can't rely on real-life scopes to align precisely with the HTTP methods, since they might be "more semantic" (perhaps amounting to specific choices of parameter usage on an endpoint?) than the endpoint division boundaries would suggest.

Some examples of scopes, from the real world and from our scenarios: The protected inbox scenario might want a scope for "can add inbox messages". Google Calendar has what amounts scopes for each calendar, in free/busy and detailed forms, which implies that the requester can only read the calendar, not write into it.

One implication of the current scope proposal is that it moves some of the responsibility for protection of resources to the host side (and possibly requiring the user to do more tasks on that side).

**AllCalendar.Com after User authentication:** Note that Domenico has assumed that AllCalendar is willing to protect each resource with different AMs. This is fine; it's up to each host whether to do this. In effect, this wireframe seems to be conflating the generic use case of introducing AllCalendar to CopMonkey and the specific use case of Alice doing stuff to her travel calendar and suddenly wanting to share it selectively. We suspect that, in reality, AllCalendar will offer only a single option of AM for everything it manages on Alice's behalf, for simplicity.

Herve observes that we have an instance of "the NASCAR problem" at every point where a host gives a user an opportunity to introduce it to one of several AMs. The NASCAR problem has abated quite a bit; the real world has shown us that IdP discovery is tractable, if you're willing to give prominence to the "big players" in the space. Herve observes that if we really have two conceptual dimensions but must squish them into a scalar scope list, we get an explosion of that list.

There is a popup that AllCalendar shows Alice before redirecting her to CopMonkey, which the host is in control of. We suspect that this popup, as currently shown, has too much policy-type information in it. By contrast, looking at the real-life Google Calendar "share this calendar" tab, it has a checkbox for "Share only my free/busy information in my calendar". That's the level of simplicity we should be shooting for. So what would Domenico's popup look like if it had only the information we think it should have? Checkboxes (or radio buttons?) in a single list with options for "Read all details", "Read only free/busy info", "Write new entries".

We're not sure we understand the Terms and Conditions. Is this meant to be a TOS for the user that governs her introduction of the host to this AM for outsourcing policy decisions?

The Privacy Impact information doesn't seem to belong here; the host might choose to show it, but maybe we should leave it out if it's not illustrating something central to UMA.
That said, the trend of having the host show the user some options and info before sending her to the AM to approve the connection is evident in the SMART project's early work on the resource registration pull model too.

The popup that CopMonkey shows Alice to get her to log in and consent should have a URL bar so that she knows she's not being phished. This could follow OAuth best practices for the consent UX. We note that the way Domenico chooses to show this popup, it includes the opportunity for the user to assign an AM-specific, "local" resource/scope string description. We think it might be better to leave this out for now, for simplicity/usability at this point in the interaction flow. If the AM wants to make this possible, this could be done elsewhere. Also, we'd like to focus very closely on UMA-protocol-specific questions for the purpose of this exercise. We do think that the AM will want to put a host "namespace" on its local version of string descriptions, to disambiguate calendars from different hosts.

What are all the user stories that we want to solve for? E.g., Alice might want to share another calendar; what should that experience look like? How would she manage the universe of “shareable resources/scopes” at that host?

**Step 3**

- Discuss Christian's proposal

Christian walked through his proposal:

1. Requester access resource using the access token
   a. the Requester uses normal OAuth but using the access token received from the AM to make the request. It's doing it as described in http://tools.ietf.org/html/draft-ietf-oauth-v2-10#page-29
   b. OAuth does not use the client_id or scope here!

2. The Host now asks the AM to verify the access using the AM Token Verification API. The Host sends access token to the Token Verification API of the AM using OAuth itself:
   b. (the first one is the one to verify, the second one is the Host's access token from the AM)

3. The AM receives this information, verifies if the access token is valid. It returns the scope it was registered for.
   a. (scope: '...') (maybe a list?)
   b. If an error occurred it will return {error: 'some error code'}
   c. (I wonder why OAuth never defines HTTP status codes instead...)

4. The Host now knows the token is valid and knows its scope, it can eventually grant access if the URL matches the scope

The host's own access token can be long-lived, or it can be short-lived and it can use a long-lived refresh token (which doesn't require user interaction) to get a fresh one as necessary.

If we can make the simplifying assumptions that the host is outsourcing token validation and the requester's access token is short-lived, Eve believes that the only big question that remains to be solved is the nature of the scope that is passed through the process. (This is still a big problem!...) Christian agrees.

Herve notes that to truly authenticate the requester, there has to be a run-time challenge for it provide information only it and the AM know. Can the requester create a nonce (a random number which is usable only once) and provide it to the AM when requesting an access token, and then provide a one-way hash of it along with wielding the access token to prove possession of it? Let's explore this.

Christian and Eve are inclined to solve for the very simplest situation, meaning host outsourcing of token validation, and then go "up" from there if necessary.

**Next Meetings**

- WG telecon on Thursday, 16 Sep 2010, at 9-10:30am PT (time chart) on line C