UMA telecon 2011-02-17

UMA telecon 2011-02-17

• Date and Time
• Agenda
• Attendees
• Minutes
  • New AI summary
  • Roll call
  • Approve minutes of 2011-02-03 and 2011-02-10 meetings
  • Newcastle University IDDY award
  • Action item review
  • Trust model review and discussion
  • Discuss reg open issues
  • Review step 2 and step 3 proposed changes
• Next Meetings

Date and Time

• WG telecon on Thursday, 17 Feb 2011, at 9-10:30am PT (time chart)
  • Skype line "C": +9900827042954214
  • US: +1-201-793-9022 | Room Code: 295-4214

Agenda

• Roll call
• Approve minutes of 2011-02-03 and 2011-02-10 meetings
• Action item review
• Trust model review and discussion
• Review step 2 and step 3 proposed changes
  • Plan to move onto OAuth 2.0 draft 12 etc. basis
• Discuss reg open issues
• AOB

Attendees

As of 8 Feb 2011, quorum is 8 of 15.

1. Mohammad, Alam
2. Catalano, Domenico
3. D’Agostino, Salvatore
4. Fletcher, George
5. Machulak, Maciej
6. Maler, Eve
7. Moren, Lukasz
8. Morrow, Susan
9. Wolniak, Maciej

Non-voting:

• Kevin Cox
• Cordny Nederkoorn
• Frank Wray

Regrets:

• Christian Scholz
• Paul Bryan

Minutes

New AI summary

| 2011-02-17-1 | Maciej et al. | Open | Examine the SMART implementation to check on the resource set identifier uniqueness implications. |
| 2011-02-17-2 | Eve | Open | Ask for John B.'s help in understanding how OpenID AB Core and the JWT work apply to our usage of the artifact and "meaningful token" design patterns. |

Roll call

Quorum was reached.
Approve minutes of 2011-02-03 and 2011-02-10 meetings

Minutes of 2011-02-03 and 2011-02-10 meetings APPROVED.

Newcastle University IDDY award

Congratulations to the SMART term on their win of this award! The OAuth Leeloo project has been moved to Apache Amber, and further work will be done under the Apache umbrella. The team will be releasing an "OAuth toolbox" shortly as well. At the ID collaboration day on Monday, Maciej M. also showed screenshots of the the UMA/j and sample application implementation work that's been going on; it will likely be released publicly in mid-March. People can go to www.smartam.net to sign up to get notifications. The team is interested in potential code contributions, even before the public release, so get in touch with Maciej if you have an interest.

(Hey, "UMA/j" kind of sounds like "Maciej"...)

Action item review

- 2010-11-18-4 Eve Open Capture new user stories in the wiki.
- 2011-01-06-4 Susan Open Write draft UMA trust model. Drafts sent to list. Slated to be closed after her meeting on Feb 21.
- 2011-01-27-1 Paul Bryan Open Revise the Claims 2.0 spec to reference and profile JWT.
- 2011-01-27-3 Paul Bryan Open Revise the "scoped access" flow list to reflect decisions made in today's meeting.
- 2011-02-10-1 Alam Open Share details of his planned UMA demo with the list.

No old AIs were closed.

The implementation work at Fraunhofer SIT has been based on top of Leeoo (so it's actually an independent implementation of the UMA piece on top of OAuth). Alam is eager to get more detail on our scoped-access design.

Trust model review and discussion

Susan recently posed key questions in email:

- Trust relationships –
  - how granular will we make them, e.g. will we separate out all possible types of trust between any given actor pair? Rather than specifying them as multiple elements of that given actor pair?
  - Should there be sub divided into trusted and untrusted phases – perhaps having these sub divisions as separate actors?
- Eve's list of TR's
- Constellations –
  - Are we happy to use constellations to model individual trust relationships within a given use case?
  - should we be applying the constellation model of teasing out trust relationships for each use case?
  - Nomenclature – are we happy to keep the UMA specific nomenclature in the model documentation (I just note this because of a specific mention of the same actor having multiple roles, e.g. the host perhaps having a dual role of IDP and RP

The intent behind "constellations" is to break down trust model detail into subsets of communication data flow. Once you've established these, you can go to individual user stories and pick out the different relationships that obtain. Eve wonders if "person-to-*" sharing profiles are the right distinctions to be made to constellations. Another dimension of distinction could be whether dynamic discovery of arbitrary parties before trust establishment is acceptable. Constellations are really just another kind of scenario/use case/user story/epic, except with a trust theme. Numbered trust relationships (TRs) are meant to be a static list, which constellations can reference and further describe.

A wise person once said that "to trust" should be substituted mentally with "is vulnerable to". It's a good test to construct sentences like this for the trust relationships and their variations, to see if it hangs together.

For now, we're agreed to use the simplest possible list of TRs (without the trusted/untrusted distinction and with only "liable actors" like the requesting party, not a requester app) and to construct "starter constellations" that use the person-to-self, person-to-person, and person-to-organization (?) distinctions found in the Legal Considerations document. We'll see how this flies in Susan's meeting with Rainer et al.

Discuss rreg open issues

We agreed that softening the requirements as Eve has suggested is a good idea.

Regarding the question about dividing resource set identifier info into section 5 vs. 6, George suggests the principle that the URI constraints be described only where the URI is created. Thus, our current division of detail is good, except that we think section 5 should have a cross-reference more explicitly to section 6.

Regarding the TODO item about requiring the resource set identifier to be unique across all users, George asks if it's possible for a resource set to be shared among multiple users. Eve believes that any one resource set will be managed by a single authorizing user at a time. Nonetheless, perhaps we don't want to over-simplify uniqueness of URIs if the host and AM can, with no problem, manage their own understandings of exactly which authorizing user is meant. There was general agreement on this point; we should just add a note warning that the host and AM have a responsibility to disambiguate through the usage of the access token.

Does this mean that we no longer have an inconvenient dependency on different resources for different users at the same host having distinct URLs, so that we could gracefully handle the "Twitter case" where the same endpoint URL is reused for everyone? Our resource set IDs are abstract – i.e., they don't map to actual URLs as far as anyone except the host is concerned – but we suspect that this subtle matter helps the situation. Resource sets then become specific to a host, and only specific to a user at that host to the extent that the host chooses to manage things this way.
Regarding the possibility that the AM might not be able to successfully grab action-set information when it needs to? Ideally we can push the whole thing off to the HTTP layer, which is what we do now. But do we want to offer “best practices” guidance? We don’t know what guidance to give yet, plus it’s very implementation-specific. The way it might work in the worked example in the reg spec is that the user’s default “tight” policy would obtain until the AM could retrieve enough action info to give the user the option to set looser policies. We agreed that maybe someday we’ll be in a position to write a separate best-practices guide, but we need more implementation and deployment experience first.

Regarding the problem of accidentally having eliminated the possibility of using “anonymous” client credentials as the host ID, the SMART project currently implements the dynamic client registration spec so that its hosts always have a unique ID. But Cordny suggests using timestamps in some clever way to ensure that each host has a unique ID. This one remains unsolved.

Review step 2 and step 3 proposed changes

Should we plan to move onto an OAuth 2.0 draft 12 etc. basis? And now draft 13 has just come out. The changes at this stage are largely editorial, and so it seems sensible to wait until the draft pace settles down a bit more.

Working in the UMA AM-host flows etherpad, we came up with this commentary:

Flow 1: independent token flow (hybrid option)

The following steps are performed:

1. The Requester gets notified that the user has granted access (is this specified? no - it has to be; this would be part of Step 2). It includes an access token for the scope (read, /photos/joe). More specifically, the access token represents the requestor’s theoretical right to seek access, and it is mapped by the AM to a list of scopes/authorizations. So, said another way, it “includes an access token that is associated with a set of scopes”.
2. The Requester accesses the resource /photos/joe (to get a listing) using OAuth, presenting the token it was just given.
3. The Host does not know about that token. It sends a verification request to the AM with that token. The “verification request” is more like a request for the AM to send it the list of scopes associated with that token, along with a time-to-live for this list. We need to specify exactly what this request/response message pair looks like.
   - How does it know it’s from that particular AM? For example, what if it’s not /photos/joe but /twitter.com/randomendpoint, not distinguishing between users? Could be a “normal” OAuth token (but then it might know that already). No problem if only one AM is attached to that users resources? This seems correct: If the URL against which access was attempted is uniquely associated with one user, there’s no problem, but if the URL is “generic”, we do have a problem. Previously we have waved our hands about this problem, saying that maybe it will encourage APIs to be more RESTful, but that’s not exactly practical.
4. The AM checks the token (which can include verifying the signature that it itself put on the token) and answers with the scope attached to that token.
5. The Host then internally maps the scopes it received from the AM to the type of access attempted and the resource on which access was attempted, and determines whether the access was on the list. It answers the request from the Requester with either “success” or with a “you are forbidden” (403?).
6. More requests are done without the AM being involved, if the time-to-live period that the AM gave was sufficiently long.
7. If the answer that the host gave the requester was “forbidden”, it includes instructions on how to go back to the AM to try and win the needed scope. Would this look exactly like the response if the requester came to the host with no token at all? How would they differ, if they need to differ at all?

Issues:

- What about invalidation of the token? Could this be handled with caching periods that are specifiable in HTTP, or do we need a method that’s in-band in the UMA protocol?
- Is short timeout enough? If the timeout is really short, the hybrid option starts looking a little more like the PDP option, where the host routinely bounces the requester over to the AM to qualify/re-qualify for the needed access authorization. We don’t need to say anything about AMs’ choices around short vs. long timeouts; they can adopt whatever model they wish according to what the performance envelopes are. The host and the requester can respond to whatever strategy the AM selects.
- How does the Host know which AM to ask?
- We might encode the scope in the token directly (e.g. via JWT). This depends entirely on JWT and its associated signature and encryption specs getting more fleshed out, and we don’t need to worry about this right now. Maybe we can get John Bradley’s help defining a sample, just to get our heads around it.
- If a lot of separate resource sets are defined then this token might get really big. We decided that if token size is a problem (once we solve for the “meaningful token” option), then we should be sure to allow AMs to choose to devolve back to the “hybrid option” gracefully.
- How do you verify this token? In the hybrid (artifact-style) option, the host is handing back to the AM a token that the AM itself signed, so the AM is doing all the verification work that it came from the “right source”. Once we solve for the “meaningful token option”, we’ll have to figure out how the host knows enough to verify a signature coming from the AM.
- Only restrictions to url and action are taken into account, not e.g. time of day restrictions etc.
- But then you might give it a short lived access token. Correct: The AM is in charge of making policy decisions and setting timeouts based on such factors.

Next Meetings

- WG telecon on Thursday, 24 Feb 2011, at 9-10:30am PT (time chart)
- WG telecon on Thursday, 3 Mar 2011, at 9-10:30am PT (time chart)
- WG telecon on Thursday, 10 Mar 2011, at 9-10:30am PT (time chart) – Eve regrets; Maciej to chair and publish agenda/minutes