UMA telecon 2016-08-18

UMA telecon 2016-08-18

Date and Time

- Thursdays, 9-10am PT
  - Skype: +9905100000481 / US +1-805-309-2350 / international lines / web calling interface / code 1782540
  - Screen sharing: [http://join.me/findthomas](http://join.me/findthomas) - NOTE: do not use the join.me dial-in line
  - UMA calendar: [http://kantarainitiative.org/confluence/display/uma/Calendar](http://kantarainitiative.org/confluence/display/uma/Calendar)

Agenda

- Roll call
- Approve minutes of UMA telecon 2016-06-30, 2016-07-07, 2016-07-14
- N.B.: No meeting next week
- For reference: MPD [sequence diagram](http://kantarainitiative.org/confluence/display/uma) and generic low-level [sequence diagram](http://kantarainitiative.org/confluence/display/uma)
- Review new editors’ draft of Core spec (forthcoming on Wednesday)
- Client awareness of scopes and set math if time
- AOB

Minutes

Roll call

Quorum was reached.

Approve minutes

Approve minutes of UMA telecon 2016-06-30, 2016-07-07, 2016-07-14: Moved by Andi, seconded by Jeffrey: APPROVED by unanimous consent.

Review new editors’ draft of Core spec

Discussion about the #155 edit on RSR: Before this edit, we just said all other operations "are unsupported", and there's an error defined for unsupported operations. But MUST NOT is very strong. Saying something like "unsupported by this specification" could be better. Or "Other operations are undefined by this specification; where interoperability by independently implemented services is desired, it is RECOMMENDED to write an extension" or something. "Undefined by this specification" or "out of scope" seems the most neutral.

Spec instructions:

- Keep this commit but change MUST NOT to "is undefined"

We can close #155 and #159.

Discussion of #153/#154:

Note that the rationale for the "saved consent token" (SCT) name is that the "requesting party token" (RPT), which is now a true OAuth access token in form/format, will also have its own regular OAuth refresh token, and the SCT thingie will have an explicit purpose to capture RpP consent/authorization, much as a traditional OAuth access token's purpose will be.

The authorization API as was is now an authorization interface that is both client- and requesting party-facing, with an OAuth extension grant called an UMA grant that starts precisely where the client first "touches" the AS. Are there any implications of HTTP's updating (the updates to RFC2616) to the UMA grant definition plans? We suspect not. Our specs don't point directly to HTTP; only OAuth; it's the OAuth token binding spec (not just RFC6750) that should take special note of the HTTP updates.

Consensus on the interface/grant approach. We have removed the UMA-specific RPT endpoint entirely, and we are overloading the existing OAuth token endpoint to produce an RPT (what we’re still calling it for the moment, though its form/format will change) and we’re giving a portion of the UMA flow around that a name: the UMA grant. The original flow was already quite similar, but took place at a special endpoint that was OAuth-protected with a scope called uma_authorization.

Saved consent token definition: The original Core Sec 1.3.2 definition of an AAT said: “An AAT binds a requesting party, a client being used by that party, and an authorization server that protects resources this client is seeking access to on this requesting party's behalf. It is not specific to any resource server or resource owner. The issuance of an AAT represents the approval of this requesting party for this client to engage with this authorization server to supply claims…” Quite a lot of that is applicable to an SCT too. Should we add that to its Terminology definition? George asks: How does the time-to-live strategy work? This goes with our DISCUSS points in the Core 00 draft's Sec 3.5.3.1. Is correlation with the RPT the correct thing to do?

(We continued informally as people started to drop off.)

The original idea behind an SCT (was "URT") was that the binding is just like AAT but happening later than the AAT point in time. It should be possible to get this token with the refresh token grant type. Could it be treated truly as a scope (the “save consent” scope) and not a token? We’re starting to get a glimmer of where Justin was going, …except that our new UMA grant needs to be much more full specified because we need to spell out:

- How the UMA client's own OAuth client credentials flow to the AS as part of this grant
- How the access token (RPT) is issued
• Whether an OAuth refresh token is issued (we think so) and its particulars
• How the SCT comes into it, and whether it's a "token" or a "ticket"-like thing

We really need a conversation with Justin next to sort out his existing implementation and where we'd like to go next in terms of spec details. We suspect that it's most like a client credentials flow, ++. The interaction endpoint on the AS would produce a consent token (ticket-like thing attached to the RPT as metadata?). Or do we need a unique consent endpoint? Can we reuse the OAuth authorization endpoint for that?? It's got the right semantic; the only problem is that it requires authentication. Our UMA-specific interaction (claims) endpoint is better designed.

Adrian notes that Michael Chen's EHR system, used by 20,000 people, has a challenge in that it needs to know which person is using the system but the hospitals need policies regarding which client is accessing the resource but doesn't want to manage groups, roles, etc. They want to give out data based on client, not by user. They want accountability at the requesting party level for break-glass use cases (Dr. Jones vs. Dr. Smith), but the policies are not specific to the requesting party. They need to audit access by the specific party. Doctors work in coverage groups.

Client awareness of scopes and set math

See 2016-06-02, 2016-06-16, and 2016-08-04 notes. Last time we said: "Let's discuss client registration for scopes more next time."

Deferred.

Attendees

As of 1 Aug 2016, quorum is 6 of 11. (Domenico, Sal, Nagesh, Andi, Robert, Paul, Maciej, Eve, Jeffrey, Mike, Sarah)

1. Domenico
2. Andi
3. Eve
4. Jeffrey
5. Sarah
6. Mike

Non-voting participants:

• Adrian
• Jin
• Kathleen
• Venkat
• Ishan
• George