UMA telecon 2018-11-02

UMA telecon 2018-11-02

Date and Time

- Thursdays 9am PT – **special day this week FRIDAY (same time as usual)**
- Screenshare and dial-in: https://global.gotomeeting.com/join/857787301
- See UMA calendar for additional details: http://kantarainitiative.org/confluence/display/uma/Calendar

Agenda

- Roll call
- Approve minutes of UMA telecon 2018-10-18
- Discuss extensions added to implementations or desired in more depth
  - Discuss Keycloak extension re claims added to permission request with Pedro
- 180 degrees use case / decoupled use case discussion
- AOB

Minutes

Roll call

Quorum was reached.

Approve minutes

- Approve minutes of UMA telecon 2018-10-18

Deferred.

Discuss extensions added to implementations or desired in more depth

Pushing claims from the RS to the AS during the permission request: Keycloak does this (using the actual UMA claim_token parameter). The use case is for when the "RS is the RO". The Keycloak likes the UMA permission ticket model because it "separates the RS from the client" and enables a human user to set sharing preferences, so this is why Pedro calls it "not the typical UMA use case".

What does the RS know that the client can't provide? What is an example of the claims it passes? The RS might want to send the company that the user belongs to, or any run-time information related to the current request for the resource. A classic example: If the client is talking to the RS via MTLS, you can send the CLN of the client cert or the cert itself. Then you can ensure the permission ticket is given only to the client that deserves it. The RS can push any information to the AS for evaluation that could free the RS from doing that job.

In the past, we’ve also talked about the client's apparent IP address when it shows up at the RS; this is one input signal that the RS could convey to the AS as part of eventual policy evaluation, which could be combined with a similar signal that the AS could detect when the (same?) client shows up at the token endpoint.

The source of claims could be anything. Keycloak provides a [Claims Information Point](http://kantarainitiative.org)) that could serve as a source. The client isn't allowed to push claims to the RS at the point of requesting a resource, though of course whatever was part of the API call the client made could be used by the RS as "fodder" in pushing information ("claims") to the AS in the permission request.

Another use case could be for when you need a specific fine-grained value constraint, akin to what Open Banking needs in order to arrange for a payment of exactly 60 pounds (rather than enabling authorization of "payment scope" generically for some app). If you’re stuck using only the OpenID Connect spec as already written, then this is why OB hunted around and leveraged the request object.

Peter notes that fine-grained authorization (scopes?) would be of interest. The primary use case is a document (may not be verified by third parties) for which temporary access needs to be granted to third parties. Customs/border control needs access for a span one hour prior one hour after my arrival. This is the “expiring entitlements” use case. They’re using consent receipts in a manner not originally intended, involving negotiation. He’s engaging with the CIS WG on this, such that the receipts can reflect the resource owner is the offerer of access. See this [blog post](http://kantarainitiative.org) and this [mind map](http://kantarainitiative.org). (The two WGs were hoping to do joint work on this – maybe we can get back to that in 2019. The UMA business model should be able to support this.)

Pedro notes that Keycloak supports claim tokens that are ID tokens, and also bearer access tokens. The HIE of One/Trustee implementation supports uPort claim tokens.

It seems there is some support for bringing some of these topics to IETF, which means engaging in a more sustained way there. Let’s discuss more next time.

180 degrees use case / decoupled use case discussion

Deferred.

Attendees

As of 18 Oct 2018, quorum is 5 of 8. (Domenico, Peter, Sal, Andi, Maciej, Eve, Mike, Cigdem)
1. Domenico
2. Peter
3. Sal
4. Andi
5. Maciej
6. Eve
7. Mike

Non-voting participants:

- Alec
- George
- Nancy
- Pedro
- Thomas
- Colin
- Bjorn

Regrets:

- Cigdem