Extending UMA protocol to Trusted Claim Approach

The UMA protocol supports the policy-driven ability of an AM to demand claims from a requesting party before authorization is granted. The claims may be self-asserted or third-party-asserted.

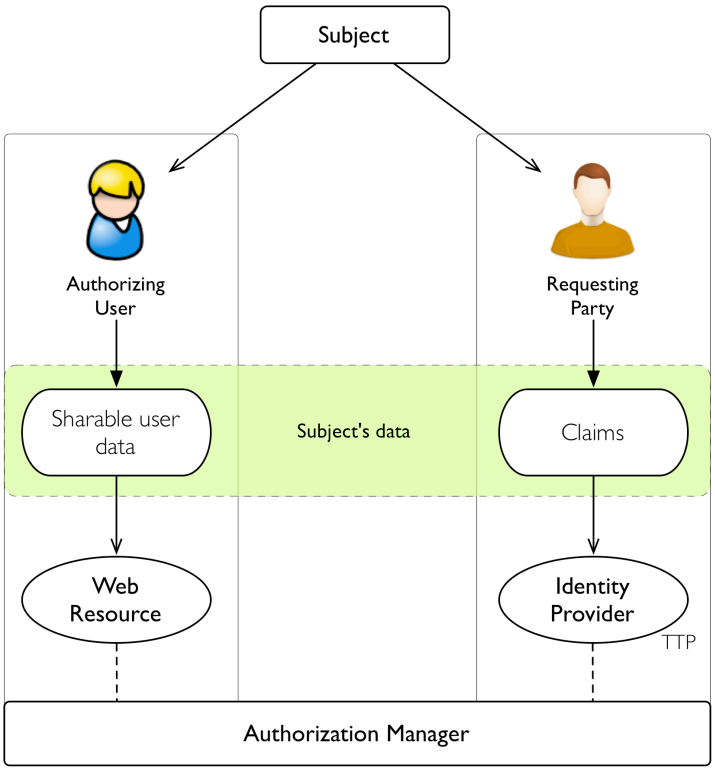
[Citation for "claims" can come from http://wiki.idcommons.net/Claim: *"An assertion made by a Claimant of the value or values of one or more Identity Attributes of a Digital Subject, typically an assertion which is disputed or in doubt."*]

When claims are self-asserted (the Claimant is the Requesting Party) and the information they represent have relatively modest needs for privacy and protection, they can be handled forthrightly by means of the simple claims request/response protocol defined in UMA's requester-AM interaction. But the power of third-party-asserted claims (where the Claimant and the Requesting Party are different), coupled with potential needs to apply higher security and privacy to claims transfer, suggests a different solution.

A typical scenario involves person-to-person data sharing, in which the Authorizing User wants to restrict sharing to a specific Requesting Party identity. For such a policy to be meaningful, such a scenario often requires that the AM trust the third-party identity claim issuer.

The proposed approach leverages the UMA protocol and introduces the concept of a Subject as a generic entity that refers to either the Authorizing User or a Requesting Party. Each Subject can perform actions on its own AM to authorize, respectively, arbitrary Web data sharing or the sharing of claims in support of its request to access another's Web data.

This model allows the creation of a comprehensive ecosystem in which the Authorization Manager can be used to protect both "classic" web resources and claim resources available from a Trusted Third Party (TTP) Identity Provider. The picture below shows how the subject's data is part of the UMA ecosystem. [I suspect we won't have room for this picture; we probably don't need it given the clarity of the text. Thoughts? -elm] [I agree. –dc]



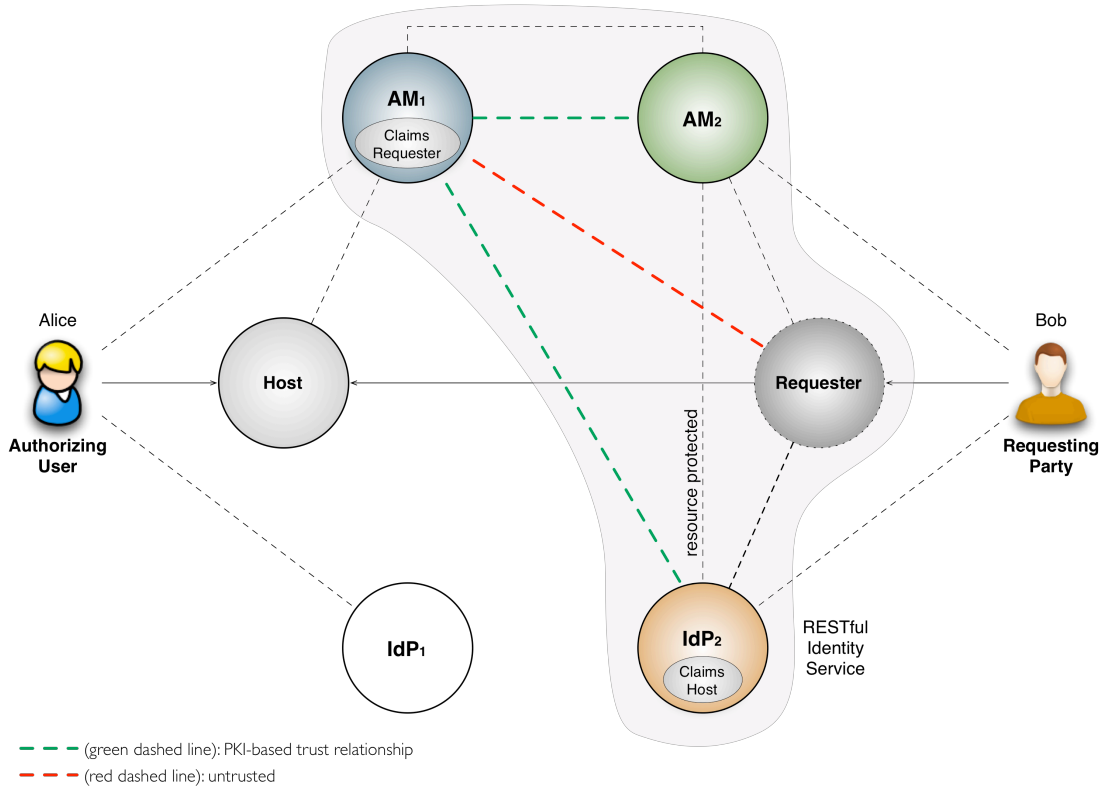
The goal is to provide a security mechanism to handle complex inter-domain trust relationships. The following picture shows the trust relationship graph that must be considered to address the scenario.

According to ITU-T X.509, Section 3.3.54, trust is defined as follows: *"Generally an entity can be said to 'trust' a second entity when the first entity makes the assumption that the second entity will behave exactly as the first entity expects."*

In our scenario, the first entity is represented by Alice's AM and the second entity is represented by Requesting Party (Bob).

Extending this model to UMA protocol terminology, we have that: [We appear to be missing IdP1 from the explanation, but in fact I'm not sure it adds much to include it, even in the diagram. -elm] [Yes, the diagram shows a light gray area which is scope of our trust investigation. IdP1 is out of scope. – dc]

- Alice's AM (AM1) acts as a Claims Requester.
- Bob acts as Authorizing User for his Identity Provider (IdP2), which acts as a Host that issues claims about him.
- Bob protects his Claims Host using an AM (AM2).



The table below shows the matrix of trust for the proposed model. [I'm not sure I understand this matrix fully; would love to find time on the phone to discuss it early this week. -elm] [I've updated the trust matrix, including more details about the meaning. I hope it works now – dc]

	Requesting Party (Bob)	**AM1** (Requester)	**AM2**	**Requester**	**Claims Host** (IdP2)
Requesting Party (Bob)	-	-	Website Authentication/SSL	AuthN	AuthN
AM1	Unknown	-	PKI-based trust relationship	Untrusted	PKI-based trust relationship
AM2	AuthN	PKI-based trust relationship	-	-	PKI-based trust relationship
Requester	AuthN	Website Authentication/SSL	Mediator	-	-
Claims Host (IdP2)	AuthN	UMA Token	UMA Introduction	-	-

The function T(A, B) describes the security mechanism by which A trusts B (respectively row and column of the matrix) and the result may be:

- Subject's Authentication-based trust (AuthN).
- Website Authentication based on SSL/TSL.
- PKI-based trust relationship (i.e. able to verify Digital signature).
- UnTrusted (no relationship).
- Unknown (there is not direct interaction between the parties).
- Mediator (redirect capabilities on behalf of the subject).
- UMA Token
- UMA Introduction

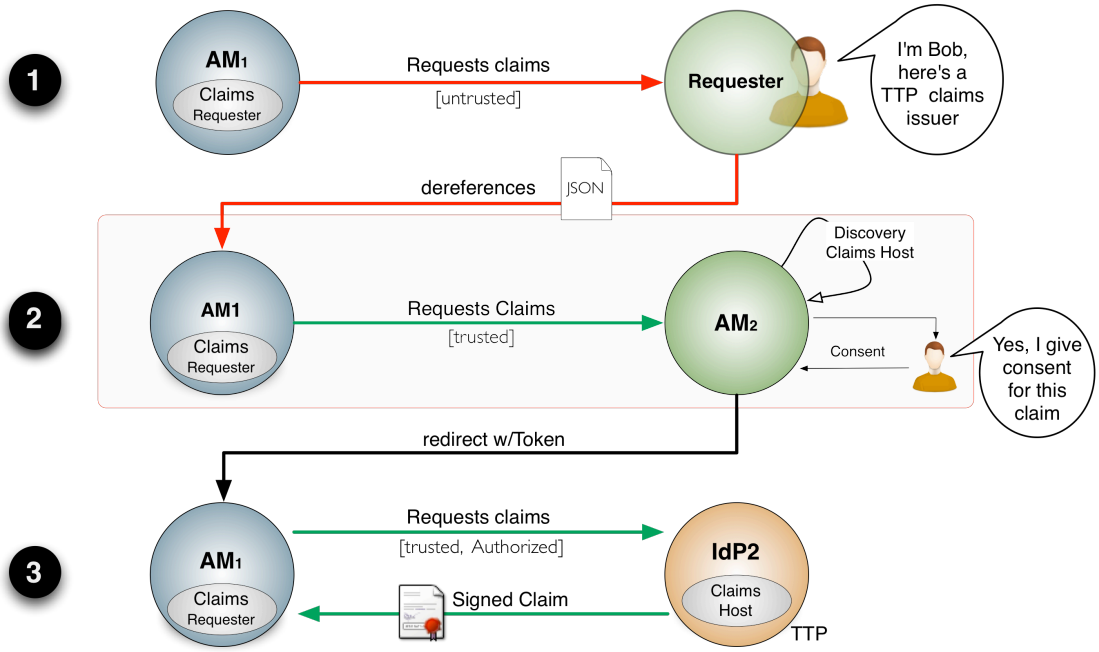
For instance, T(AM1, IdP2): *PKI-based trust relationship*, this means that AM1 is able

to verify and trust IdP2's digital signature.

Based on the previous matrix, we are able to prove how the propagation of trust works when AM1 demand claims from a requester before authorization is granted.

1. *T(AM1, Requester)* - Based on Authorizing User policy, AM1 requires claims from the Requester. The Requester, on behalf of the Requesting Party, redirects AM1 to the subject's Authorization Manager (AM2), which is in charge the sharing of claims. There is no trust relationship between AM1 and the Requester.
2. *T(AM1, AM2)* - AM1 requests claims to AM2. AM2 discovery the claims Host and requests consent from the Claim's Authorizing User before to redirect AM1 to Identity Provider (IdP2). There is a direct trust relationship between AM1 and AM2. [Note that this is a profile/specialization of core UMA; we are dictating that Bob must have a real-time role to play in authenticating and authorizing the sharing of claims to support his resource request. Right? -elm] [Absolutely, Bob has a real-time role. However, AM2 could offer default policy for claims sharing that Bob can select to avoid his direct involvement in the process, this can depend on the sensitivity of the claims. In this case, AM2 can provide notification to Bob (e.g. SMS), in order to mitigate risks –dc].
3. *T(AM1, IdP2)* - AM1 requests the claims from IdP2 using an access token.

Finally, at the end of the process, AM1 gets the signed claims from the IdP2.



Identity Assurance feature

The trust model leverages the notion of a Trust Framework [Do we need to provide a citation here to a definition of this? How about the OITF paper? (See info here: http://www.xmlgrrl.com/blog/publications/#oitf) -elm] [I agree, OITF provides a great profile about the identity assurance problem, including references to LOA,

ICAM and NIST -dc] with the following assumptions:

- AMs register with a Trust Framework Provider (TFP) to receive notification about new IdP Claim Issuers.
- IdPs must apply to TFP to receive a Level Of Assurance (LOA) certification. (Step. 1.0)
- AMs will be notified about new Claim Issuers certified by the TFP. (Step. 2.0).
- AMs get the IdP list updated from the TFP (including cryptographic elements). (Step 2.1)
- AMs provide to their Subject a certified IdP list (with LOA certificate).
- The Subject can select a preferred IdP (Claim Issuer) to initialize an introduction process with the AM. (Step 3.0 and 4.0)
- The IdP (Claim Issuer) becomes a trusted and protected resource by the AM. (Step 5.0)