User Stories

User Stories

These user stories are designed to evoke the benefits and value of various UMA protocol features, paint a fuller picture of potential user experiences, and highlight security needs. Rows in which epics (tightly bound collections of stories) are defined have the **epic title in bold**. Rows in which regular individual stories are defined have the **story title in bold**. Rows that have the *story title in italic* are "negative" user stories, in which a malicious party is seeking to do something that that must be avoided; in these cases the "How to measure" column is stated as a mitigation of the risk.

TODOs/issues:

- Access sought by the requesting user (person-to-person sharing), requesting entity (person-to-service sharing), requesting entity rep (person-to-service sharing with UX needed on the requesting side), and authorizing user as requesting user (person-to-self sharing)
- Assign persistent numbers to the stories, in addition to auto-numbering sorted versions of the rows?
- Negative story: malicious host correlating same user's activities across hosts (related to DP9, R3)
- Gather promises/claims stories into claims epic: share selectively based on dynamically provided characteristics of requesting parting (stories: AM requests claims based on policy; requester conveys claims on requesting party's behalf; user manages sets of characteristics/criteria, including ACLs of identities; optional Claims 2.0 stuff...) – related to R0b
- Epic for accessing resource if authorized (stories: requester presents access token; move negative story about fraudulent access here; host validates token with AM's help; liability concerns)
- Add trusted claims story in post-1.0 backlog?
- Add story about resource baskets/grouping of scoped resources at AM?

Epic title	Story title	*as a (n)...*	*I want to...*	*so that...*	Category	Backlog	Depends on	How to measure	Comments
Introduce host and AM	(epic)	As an authorizing user...	I want to tell each of my hosts which AM I want it to use...	so that I can control the selective sharing of resources at a variety of my hosts from one place.	UX	1.0		User can choose an arbitrary AM dynamically (optional), or can select among whichever preconfigured AMs the host finds acceptable.	R1
Introduce host and AM	**Discover AM metadata**	As a host...	I want to discover an AM's UMA endpoints dynamically...	so that I can begin offering resource protection services to my user using whatever AM they prefer.	Protocol	1.0		Host uses AM address to construct and dereference a hostmeta address. AM hostmeta contains sufficient UMA endpoint data to get started.	Core spec step 1
Introduce host and AM	**Obtain user authorization for host-AM introduction**	As a host...	I want to obtain my user's consent to use their chosen AM for resource protection...	so that I can interact with the AM securely on behalf of my user in offering this protection.	Protocol	1.0	Get client credentials as host (story)	Host gets host access token and optional refresh token from AM through user authorization using the OAuth 2.0 web server profile.	Core spec step 1
Authenticate to AM as a client	(epic)	As an AM...	I want to securely distinguish each host and requester...	so that I can track their interactions with me, and enable my authorizing user to track such interactions, over time.	Protocol	1.0		AM can keep an accurate record of which host or requester has approached it in each instance, correlating by client ID.	Dyn reg spec
Authenticate to AM as a client	**Issue client credentials**	As an AM...	I want to assign unique OAuth client IDs and optional secrets to each host and requester...	so that I can uniquely and securely distinguish individual hosts and requesters when they approach me on behalf of any of their respective users.	Protocol	1.0		Host or requester either statically acquires a client ID and optional secret (out of band), or dynamically acquires them from an AM endpoint meant for this purpose.	Dyn reg spec
Authenticate to AM as a client	**Require client identification and authentication**	As an AM...	I want to securely identify and authenticate hosts and requesters that approach me for an access token or other interaction...	so that I can track interactions with them accurately.	Security	1.0		Host or requester presents client ID and secret (when one has been issued) to AM when interacting with it. Host or requester does this only over a protected channel such as SSL.	Core spec step 1 (host), step 2 (requester)
Authenticate to AM as a client	*Impersonate a client*	As a malicious entity...	I want to fraudulently obtain and use a legitimate host's or requester's client credentials at an AM...	so that I can impersonate it in interacting with that AM, hopefully leading to fraudulent access authorization or other malicious behavior.	Security	1.0		AM issues a client secret in cases where client authentication is important (out of band in static registration cases). Host or requester acquires client secret only over a protected channel such as SSL (out of band in static registration cases). Host or requester protects client secret at rest (out of band). Host or requester presents client ID and secret (when one has been issued) to AM only over a protected channel such as SSL when interacting with it.	Core spec step 1 (host), step 2 (requester); dyn reg spec

Share resources selectively	(epic)	As an authorizing user...	I want to set up selective sharing of one or a set of resources residing at any of my hosts on the web...	so that I can ensure the resources are shared only with parties I choose, and only in ways I choose, allowing me to track sharing using a single "hub".	UX	1.0	Introduce host and AM (epic)	User can select desired resources, scoping, operative policies and terms, and any requesting-party constraints in some fashion. User can track sharing and policy details from one place.	R0a
Share resources selectively	**AM-manage resource without sharing**	As an authorizing user...	I want to indicate that a scoped resource is entirely "hidden" from view...	so that I can protect it without worrying about unauthorized access, and decide later at my leisure who to give access to.	UX	1.0		User can put a host's resource under AM protection/management in such a way as to attach a "do not share" policy to it. User can track authorization attempts through the AM. User can later attach a different policy that allows access authorizations.	R0b
Share resources selectively	**AM-manage resource without protection**	As an authorizing user...	I want to indicate that a scoped resource is AM-managed but "public"...	so that I can gain the ability to track access from a single "hub" without having to constrain access, and easily change my mind about constraining access later.	UX	1.0		User can put a host's resource under AM protection/management in such a way as to attach a "share with all" policy to it. User can track authorization events through the AM. User can later attach a different policy that allows different or fewer access authorizations.	R0b
Share resources selectively	**Register scopes**	As a host...	I want to convey to an AM that my user wants it to protect access to one or more scopes of access to the resources I host on the user's behalf...	so that I can offer sophisticated access control features to my user without having to perform the complexities of access authorization myself.	Protocol	1.0		Host registers scope details at AM, presensting host access token to do so. AM retains a correct representation of these details. AM can map policies to host-specific scopes on the authorizing user's instructions.	R4, resource reg proposal
Share resources selectively	**Check which scopes are registered**	As a host...	I want to get confirmation from an AM which scopes it thinks are registered on behalf of one of my users...	so that I can mitigate the risks of getting out of synchronization with the AM.	Protocol	1.0	Register scopes (story)	Host can retrieve currently registered scopes on a user's behalf. Host can subsequently register scopes as a corrective action.	Resource reg proposal
Share resources selectively	**Request registration of resources and available scopes**	As an AM...	I want to request resource registration details from a host directly...	so that any changes the user has made to resources at the host since the last time they visited the AM will be automatically picked up for policy mapping here.	Protocol	Pending		User changes to resources (such as deletion of resources or addition of new resources intended to be protected) performed solely by interaction with the host are reflected in AM's representation of which resources at that host are protected with which policies. AM can automatically attach policies to resources that did not exist when user originally directed AM to attach policies to related resources at the same host.	No consensus to solve this in 1.0
Share resources selectively	**Attach a policy to several resources at multiple hosts**	As an authorizing user...	I want to selectively share several scoped resources, possibly residing at multiple hosts, under the same policy regime...	so that I can unify my management and monitoring of access authorization of all of the resources as a set.	UX	1.0		AM associates chosen policy with multiple scoped resources. Access authorization to any of the resources is granted to the same set of requesting parties under the same conditions, such as presenting the same claims (by value) in satisfying identical claims requests.	
Share resources selectively	**Set up selective sharing efficiently**	As an authorizing user...	I want to indicate how I want to share a resource solely while visiting the host of that resource...	so that I can share resources in the most efficient and friction-free way possible.	UX	Pending		Host maps selected resources to scopes in some AM-independent way. Host conveys to AM only the scopes under that AM's protection, not specific resource knowledge.	Originally known as "Problem B"
Share resources selectively	**Change policy**	As an authorizing user...	I want to modify the policy that applies to a scoped resource...	so that I can ensure that access to the resource is exactly as broad or narrow as I wish as my needs change.	UX	1.0		AM provides a feature for user to modify policies and map different policies to scoped resources.	R0c
Share resources selectively	**Allow user to change policy**	As an AM...	I want to allow my user to modify the policy that applies to a scoped resource I manage...	so that I can be responsive in offering the level of selective sharing the user wants.	Protocol	1.0		AM changes the criteria for issuing access and refresh tokens as soon as the policy is changed. AM invalidates all existing refresh tokens and requires requesters to re-qualify. (@@correct? what about scope upgrades etc.?)	
Share resources selectively	**Stop access to resources**	As an authorizing user...	I want to stop further access to one or more of my resources...	so that I can remediate access problems, protect resources that suddenly become sensitive, or choose to become a more private person.	UX	1.0		AM provides a feature for user to revoke authorization to specific requesting parties immediately.	R0c
Share resources selectively	**Allow user to stop access to resources**	As an AM...	I want to allow my user to stop further access to one or more scoped resources I manage...	so that I can be responsive in offering the level of selective sharing the user wants.	Protocol	1.0		AM invalidates all existing refresh tokens. (@@correct? enough?)	

–	**Extract promises**	As an authorizing user...	I want to associate required promises with a policy applying to a scoped resource...	so that I can extract enforceable promises from requesting parties regarding their access to that resource.	UX	1.0		User can set up one or more promises as required claims associated with a policy that gets associated with a resource.	
	Require promissory claims	As an AM...	I want to require promissory claims to be conveyed by a requester from a requesting party...	so that I can authorize access only for requesting parties that meet my authorizing user's requirement for promises.	Protocol	1.0		AM generates only the claims-required messages that match the user's policy instructions and correctly assesses the status (sufficient or insufficient) of any claims returned in response.	
–	**Reserve authorization management to the host**	As a host...	I want to control which of a user's resources are available for AM protection /management or not...	so that, for liability, legacy, or practicality reasons, I can retain control over some portion of access authorization management over those resources.	UX	1.0	Share resources selectively (epic)	Host can unilaterally discriminate between AM-manageable resources and non-AM-manageable resources (out of band).	
–	**Authenticate to AM as user**	As an AM...	I want to authenticate my authorizing user...	so that this user's interactions with me are kept private and secure.	UX	1.0		The correct user can access their own AM settings and preferences. Other users can't access that user's settings and preferences.	
–	*Obtain fraudulent access*	As a malicious requester...	I want to fraudulently obtain an access token meant for a legitimate requester and use it at a host...	so that I can gain access to a protected resource to which I do not have rightful access authorization.	Security	1.0		Requester to which an access token has been issued is correlated with requester which uses that access token so that a different requester can't successfully use it.	

Template

If you edit the table above, you can copy and use the following template to start new rows.

(epic name)	(title)	Authz user/AM/Host/Requester/Requesting user/Requesting entity /Requesting entity rep/Malicious	I want to	so that	Protocol /Security/UX	1.0/1.0 optional/Post-1.0/Never /Pending/(none)	(dependencies)	(metrics)	(comments)