UMA’s relationship to distributed authorization concepts

19 October 2013

tinyurl.com/umawg

@UMAWG
Basic terms and concepts in UMA

- **RO** has discretionary access control rights over protected resources
- **RS** is capable of being a full PEP
  - Responsible for the “grain” of protected resource access
- Interface is standardized in UMA
  - AS and RS are capable of being fully loosely coupled and multitenant
- Client conveys an access token (requesting party token or RPT) to RS in the OAuth fashion
- **AS** is capable of being a full PDP
  - Responsible for policy characteristics (thus implicitly a PAP and PIP)
- **RqP** and **client** together constitute an access requester
  - Require an access decision by some combination of AS and RS
Implications of AS-RS loose coupling and multitenancy

- **Manage resources**
  - RO1 uses AS1 to protect resources from more than one RS

- **Protect resources**
  - RO2 uses more than one AS to protect its various resources

- **Administer policies**
  - RS operators and AS operators may all be different parties
  - May need to operate at Internet scale, with mobile and other devices
Opportunities for distribution of decision-making responsibility in UMA

- Balance depends on nature of authorization data associated with RPT
Default distribution of decision-making responsibility in UMA

- Mandatory-to-implement **bearer** RPT profile gives AS *most* power because AS and RS operators may differ
- Conveys current permissions and requires RS to match to access attempt

- Many other distribution schemes are possible through RPT profiling
- Authorization data can be contained in RPT or retrieved by RPT introspection ("bearer" profile requires the latter)
Some additional options for decision-making responsibility

- **XACML-like (AS 100%)**: RPT conveys Permit/Deny decision
- **SSO-like (AS 0%)**: RPT conveys only claims gathered from the user
- **Sticky policy-like (AS <50%)**: RPT conveys operative policies in some format
Basic elements of policy expression and their authoritative sources

- AS is exclusively authoritative for constraints on which access requesters can gain access, binding subj to verb/obj
- AS and RO together are responsible for administering all relevant policies
- RS is exclusively authoritative for protected resource sets (obj) and scopes of operation over them (verb)
- It registers them with AS through a resource set registration API

Without UMA, OAuth on its own only handles scopes; “resource sets” are implicit (come from API developer documentation)
Degree of interoperability of policy expression elements

- Policy expression and evaluation are performed outside UMA
- AS can serve as a client of other standard or proprietary PDPs, PIPs, and PAPs

- Resource set registration API and format (modular spec called by UMA) enable interop and standardization of resource set types and of scopes

- XACML and several other specifications fully standardize all elements

- OAuth scopes are specific to the API being protected; their expression is not standardized
Final observations

administer policies

AS1

(RO1) (RO2)

AS2

(RO1) (RO2)

• Not standardized by UMA
• UMA can integrate with other solutions (XACML standard, procedural code...)
• Each AS operator may choose a different solution (impacting multi-AS interop for one RO or RS)

protect resources

RS1

(RO1) (RO2)

RS2

(RO1) (RO2)

• Expression standardized by UMA
• Each RS operator can define its own resource sets and scopes, or use ones defined by others

manage resources

RO1

(RO1)

RO2

(RO2)

• Each service has its own proprietary (or standard) API