UMA Release Notes

Abstract

This document contains non-normative release notes produced by the User-Managed Access Work Group explaining how new versions of the UMA specifications differ from previous ones.

Status

This document includes release notes for all versions of UMA.

Editor

- Eve Maler

Table of Contents

Introduction

This document contains non-normative release notes produced by the User-Managed Access Work Group explaining how new versions of the UMA specifications differ from previous ones.

NOTE: Reading the release notes is not a substitute for reading the specifications carefully. In each specification release, much work is typically done to improve clarity and applicability for implementers and others. See the UMA Implementer's Guide for additional commentary.

The UMA specifications use Semantic Versioning:

> *Given a version number MAJOR.MINOR.PATCH, increment the:*
>
> 1. *MAJOR version when you make incompatible API changes,*
> 2. *MINOR version when you add functionality in a backwards-compatible manner, and*
> 3. *PATCH version when you make backwards-compatible bug fixes.*

The following shorthand terms and abbreviations are used in this document (see also the terminology, including abbreviations, defined in the specifications):

- AS: authorization server
- RS: resource server
- Core: UMA Core specification (applies to versions 1.0 and 1.0.1)
- RSR: OAuth Resource Set Registration specification (applies to versions 1.0 and 1.0.1)
- Grant: UMA Grant for OAuth Authorization (applies to version 2.0)
- FedAuthz: Federated Authorization for UMA (applies to version 2.0)
- I-D: IETF Internet-Draft specification
- Sec: section

Where a change relates to a GitHub issue, the linked issue number is provided.

From UMA1 to UMA V2.0

The UMA V2.0 Recommendations are User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization (known as "Grant") and Federated Authorization for User-Managed Access (UMA) 2.0 (known as "FedAuthz"). The official versions are downloadable from the Kantara Reports & Recommendations page; this document links to specific sections within the HTML versions.

Differences and changes noted are between V2.0 and V1.0.*n* generally; note that internal revision differences between UMA2 revisions are not tracked here. (You may find it helpful to refer to the Disposition of Comments document, a record of specification changes during the Public Comment periods late in their final review cycle, and the GitHub repository where the specifications are managed.) Where the distinction between V1.0 and V1.0.1 is important, it will be noted; otherwise the label "UMA1" is used.

The following sequence diagrams may be of assistance as brief summaries of changes made:

- Sequence diagram for Grant, highlighting key changes from UMA1
- Sequence diagram for FedAuthz, highlighting key changes from UMA1

Version Themes

The major themes of this version, as determined by the Work Group's 2016 roadmap planning process, were (along with constantly improving security) to:

- Increase OAuth 2.0 alignment
- Improve Internet of Things readiness
- Improve readiness for "wide ecosystems", where the requesting party and the resource owner's AS have no pre-established relationship

Specification Reorganization and Conformance Levels

The two specifications were divided differently until late April 2017. Core and RSR were recombined into Grant and FedAuthz, as follows:

- All communications of the client and requesting party with the AS appear in Grant. This specification formally defines an extension OAuth grant.
- All communications of the resource owner and resource server with the AS appear in FedAuthz. This includes:
    - Policy setting (outside the scope of UMA)
    - PAT definition and issuance
    - Protection API
        - Resource registration (previously, RSR specified only this endpoint/API and Core specified everything else)
        - The RS's permission requests at the AS
        - The RS's token introspection at the AS
- The formal profiles for API extensibility URIs `https://docs.kantarainitiative.org/uma/profiles/prot-ext-1.0`, `https://docs.kantarainitiative.org/uma/profiles/authz-ext-1.0`, and `https://docs.kantarainitiative.org/uma/profiles/rsrc-ext-1.0` were removed and replaced with recommendations (Grant Sec 4 and FedAuthz Sec 1.3) to define profiles as needed and to use `uma_profiles_supported` metadata (Grant Sec 2) to declare them.

It is now optional to implement the features appearing in FedAuthz; thus, this specification effectively defines a conformance level. (Note: To receive the full benefits of "user-managed access", it is best to implement and use the features of both specifications.)

Terminology Changes

Note the following terminology changes made throughout the specifications. (256) See also Summary of API and Endpoint Changes below for naming changes made to some of the endpoints.

UMA1	UMA2	Comments
configuration data	metadata, discovery document	For better clarity and OAuth alignment
policies	authorization grant rules, policy conditions	For better consistency
protection API token (PAT)	protection API access token (PAT)	For better clarity and OAuth alignment
resource set, resource set registration	resource, resource registration (protected while registered)	For better clarity and OAuth alignment
authorization API	UMA grant (an extension OAuth grant)	Result of redesign (see Token Endpoint Replaces RPT Endpoint; Client-Side Communications Defined as Extension Grant)
authorization API token (AAT)	goes away; a new related token is persisted claims token (PCT)	Result of redesign (see AAT Removed in Favor of PCT)
register a permission (for permission ticket)	request (one or more) permission(s) (on behalf of a client)	For better clarity
trust elevation	authorization process and authorization assessment	Result of redesign (see Authorization Assessment Gains Precision)
claims pushing + claims gathering = (n/a)	claims pushing + claims gathering = claims collection	For better consistency
step-up authentication	(n/a); just authorization process	Result of redesign (see AAT Removed in Favor of PCT and Authorization Assessment Gains Precision)
RPT as an UMA access token	RPT as an OAuth access token	Result of redesign (see Token Endpoint Replaces RPT Endpoint; Client-Side Communications Defined as Extension Grant)

API and Endpoint Changes

These design changes include naming changes made to some of the endpoints.

UMA1	UMA2	Comments

`.well-known/uma-configuration`	`.well-known/uma2-configuration`	The same authorization server can have two different discovery endpoints, one serving UMA1 metadata and one serving UMA2 metadata.
OAuth endpoints: • authorization endpoint • token endpoint	OAuth endpoints: • authorization endpoint • token endpoint	Previously, the token endpoint issued both PATs and AATs. Now the token endpoint issues PATs and RPTs; there are no AATs. (Note that the authorization endpoint is used for authenticating resource owners only, not requesting parties.)
Protection API: • resource set registration endpoint/API • permission registration endpoint • token introspection endpoint	Protection API (now OPTIONAL): • resource registration endpoint/API • permission endpoint • token introspection endpoint	In the case of the first two endpoints, there are both design (primarily syntax) and naming differences, which also affects their corresponding metadata in the authorization server discovery document.
Authorization API: • RPT endpoint	-	In UMA2, there is no authorization API. The prior function of the RPT endpoint is served by the existing OAuth token endpoint.
Requesting party claims endpoint	Claims interaction endpoint	This is just a naming difference.

Authorization Server Discovery Document and Metadata Changes

Discovery Document and Metadata Simplification

UMA1's endpoint and feature discovery mechanism was defined in total by its Core specification. UMA2 makes use of the OAuth Authorization Server Discovery mechanism instead (still in Internet-Draft form at the time of UMA2 publication), eliminating metadata fields already defined by the OAuth discovery or OpenID Connect specification. The Grant (Sec 2) and FedAuthz (Sec 2) specifications each define only the metadata fields they require. (59, 157, 159, 305)

Definition of OAuth Dynamic Client Registration Metadata Field

The new metadata field `claims_redirect_uris` enables the client to pre-register claims redirection URIs. (Grant Sec 2, Sec 3.3.2, Sec 7.3) (337 sub-issues c and d)

permissions Claim and Sub-Claims in Token Introspection Object Not Requested to Be IANA-Registered as JWT Claims

Previously, it was intended to make an IANA registration request of the claims inside the introspection object as independent JWT claims. This would enable them to be formally used in RPTs, such that an RS can validate the access token locally with these claims packed inside it. Because of potential security and privacy considerations, it was determined not to define this token format for now. (FedAuthz Sec 9) (334)

Changes to AS-Client, RS-Client, and AS-Requesting Party Interfaces (Now UMA Grant Specification)

Authorization Server Rotates Permission Ticket

After the AS initially generates the permission ticket and the RS conveys it to the client, whenever the client subsequently approaches the AS token endpoint or redirects the requesting party to the AS claims gathering endpoint, the AS is required to rotate the value of the permission ticket every time it hands a permission ticket value back to the client (Grant Sec 3.3.3, Sec 3.3.6). This action obsoletes the need for the UMA Claims-Gathering Extension for Enhanced Security specification (see this explanation of that specification for more information).

Token Endpoint Replaces RPT Endpoint; Client-Side Communications Defined as Extension Grant

The specialized RPT endpoint was removed in favor of using the standard OAuth token endpoint (Grant Sec 3.3.1). A formal extension OAuth grant was defined (same section), working with regular OAuth capabilities and OAuth error codes to the extent possible (Sec 3.3.6). This enabled reuse of large portions of the threat model and the client type model, along with the ability for the client to request scopes and to authenticate using its own client credentials at the token endpoint (see the next section for additional discussion). (153, 165)

AAT Removed in Favor of PCT

An end-user requesting party no longer needs to mediate issuance of an AAT at the AS, and the client no longer needs to use an AAT in order to request a token; it simply uses its own client credentials at the OAuth token endpoint as in a normal grant (see Token Endpoint Replaces RPT Endpoint and Client-Side Communications Defined as Grant). Thus, the first time the requesting party needs to interact with the AS, if at all, is to provide claims interactively when redirected by the client as part of claims collection. This is in contrast to UMA1, where an end-user requesting party would have been expected to engage in an interactive OAuth flow to log in and then authorize AAT issuance at the AS's authorization endpoint. In UMA1, the (required) AAT could have been used by the AS as a reminder of claims about the current requesting party. In UMA2, the (optional) PCT is available to serve in this capacity instead, without the OAuth mechanism being involved (Grant Sec 3.3.1). Note that UMA2 does not require the AS to involve the requesting party in an interactive flow authorizing PCT issuance (Grant Sec 3.3.3). (154, 264)

Deprecated Response-Body Permission Ticket Return Option By RS Removed

In UMA V1.0.1 the RS was able to return the initial permission ticket to the client in the response body for backwards compatibility with UMA V1.0, but this option was deprecated; now this option has been removed. (233)

Permission Ticket Return By AS With Redirect-User Hint No Longer Deprecated

In UMA V1.0.1 the AS was able to return the permission ticket to the client along with the `redirect_user` hint, but the client was not supposed to depend on ticket accuracy, and the supply of this ticket was deprecated. Now all permission tickets directly supplied by the AS are rotated and the value is safe for the client to depend on (Grant Sec 3.3.6). (233)

More Discretionary Permission Requests

The instruction for the RS to request permissions on the client's behalf (which can be a private interface or the standardized interface governed by FedAuthz) is now defined as a recommendation ("SHOULD") to be reasonable for the client's resource request, rather than being required to meet it ("minimally suffices"). The UMA Implementer's Guide has a section on Considerations Regarding Resource Server Permission Requests that explains how and why this level of discretion is more appropriate.

need_info Response Structured Flattened

The JSON nested object structure of the `need_info` error response from the AS has been flattened. Now it directly contains a permission ticket and either a `required_claims` or a `redirect_user` hint (or both) (Grant Sec 3.3.6). (237, 308)

not_authorized Error Renamed to request_denied

The UMA1 error `not_authorized` has been renamed to `request_denied`. Note that this error was re-added only in a later revision of UMA2. See the UMA Implementer's Guide section called Understanding Authorization Server Response Options From the Token Endpoint to understand AS error semantics. (Grant Sec 3.3.6) (340)

Added interval parameter to request_submitted Error

An optional `interval` parameter was added to the `request_submitted` error to enable the AS to inform the client about appropriate polling intervals. (Grant Sec 3.3.6) (341)

New Refresh Token Clarity

It has been clarified that the AS can issue a refresh token and the client can use the refresh token grant to attempt to get a new RPT with it (Grant Sec 3.3.5, Sec 3.6). (238, 284)

Authorization Assessment Gains Precision

Inputs to authorization assessment and results calculation are more normative and precise. It is also now possible for permissions with zero scopes to be granted (Grant Sec 3.3.4). (266, 310, 317)

Permission Ticket Ecosystem Rationalized

The permission ticket generation ecosystem has been rationalized. In UMA2, a permission ticket is always generated, and the value rotated, in cases of a redirect back from the claims interaction endpoint and in cases of `need_info` and `request_submitted` errors from token endpoint requests, and never in cases of other errors. An authorization process is still ongoing while the authorization server is still generating permission tickets. (275, 279, 298)

Only One Pushed Claim Token Now Allowed at a Time

In UMA1, the mechanism for claim token pushing was a JSON-encoded request message sent to the RPT endpoint, optionally including with a `claim_tokens` array each of whose objects had a `format` parameter and a `token` parameter. In UMA2 (Grant Sec 3.3.4), , due to increased alignment with OAuth, this structure was flattened and the request message – now sent to the token endpoint as `application/x-www-form-urlencoded` format – contains each of the inner parameters only once. (If it is desired to send multiple claim tokens in a single request message, a compound claim token format could be defined.)

RPT Upgrading Logic Improved

UMA2 includes more comprehensive and normative logic around RPT upgrading (Grant Sec 3.3.5, Sec 3.3.5.1). (281)

Token Revocation Clarifications

UMA2 includes more comprehensive and normative text around token revocation, and defines a token type hint for PCTs (Grant Sec 3.7). (295)

Refresh Token Grant and Downscoping Logic Clarifications

UMA2 ensures that the logic of downscoping during token refreshing is properly defined given that UMA scopes are bound to resources, and clarifies that the AS does not perform authorization assessment in this context (Grant Sec 3.6). (306)

Changes to AS-RS Interface/Protection API (Now Federated Authorization Specification)

Resource Registration Endpoint

Extraneous URL Parts Removed From Resource Registration API

The API available at the resource registration endpoint required the path to contain the string `resource_set`. This string has ben removed (FedAuthz Sec 3.2). (155)

Scope Description Documents No Longer Expected to Resolve at Run Time When Scopes Are URLs

The AS is no longer expected to resolve scope description details at resource registration time or at any other run-time requirement (FedAuthz Sec 3.1.1). (269)

Resource Descriptions Lose uri Parameter

The `uri` parameter in the resource description was removed due to potential security and privacy concerns. (FedAuthz Sec 3.1) (270)

Resource and Scope Description Documents Gain Description Parameters

Resource description documents and scope description documents each now have a new parameter, `description`, for a human-readable string describing the resource or scope (respectively) at length. (271, 272)

scopes Parameter in Resource Description Document Renamed to resource_scopes

The `scopes` parameter in the resource description document has been renamed to `resource_scopes` (FedAuthz Sec 3.1). (318)

New HTTP 400 and invalid_request Error

For a typical variety of malformed-request errors, a response of an HTTP 400 (Bad Request) status code and an optional `invalid_request` error code is now defined. (FedAuthz Sec 3.2) (354-1)

Permission Endpoint

Requesting Multiple Permissions and Permissions With Zero Scopes

It is now possible for the RS to request multiple permissions on the client's behalf, not just one; this enables the RS to request "packages" of multiple resources that are likely to need to be accessed together. It is also possible for the RS to supply zero scopes on a requested permission (FedAuthz Sec 4.1); this is because the client can request its own scopes directly from the AS (for more discussion see Token Endpoint Replaces RPT Endpoint; Client-Side Communications Defined as Extension Grant). (317)

Token Introspection Endpoint

scopes parameter renamed to resource_scopes in Introspection Response Object

The `scopes` parameter in the token introspection response object has been renamed to `resource_scopes` (FedAuthz Sec 5.1.1). (158)

Options Not to Use Token Introspection Explicitly Allowed

In UMA2, the RPT is explicitly a type of OAuth access token, and it has been clarified that the token can be self-contained and valided locally by the RS, or introspected at the AS at run time, or its cached value used as appropriate (FedAuthz Sec 5). (261)

permissions Claim in Token Introspection Object Must Be Used

If token introspection is used (see Options Not to Use Token Introspection Explicitly Allowed), the introspection object can no longer be extended to replace the `permissions` claim with an entirely different structure. (322)

permission Claim exp Sub-Claim's Meaning If Absent Removed

The statement about the `permission` claim's `exp` sub-claim not expiring if it is absent was removed for the multi-part rationale given in the linked issue. (337 sub-issue a)

From V1.0 to V1.0.1

The UMA V1.0 specifications (Core, RSR) were approved in March 2015. The UMA V1.0.1 specifications (Core, RSR) are were approved in an All-Member Ballot to be Kantara Recommendations and were published in December 2015.

The following release notes are catalogued according to their impact on software implementations (where impact on client software in addition to authorization server or resource server software is denoted with **(+Client)** in the section title). Links to relevant GitHub issues and specific section numbers are provided where possible, enabling old-to-new text comparisons and tracking of discussions and rationales.

The following themes animated the V1.0.1 release process:

- Account for V1.0 lessons learned out of the gate
- Achieve timeline predictability and minimization of disruption for V1.0 implementers
- Achieve efficiency, speed, and accuracy in specification revisions
- Achieve issue solution consistency with OAuth 2.0 and OpenID Connect where possible
- Within the allotted time, prioritize first blocking and critical bug fixes, then low-impact specification and implementation changes

Minor changes, such as changes that don't impact implementations or specification interpretations, are not discussed in this section. To see a full list of issues disposed of and specification commits related to V1.0.1, see the list of GitHub issues with the "V1.0.1" label and the commit histories for Core and RSR.

Changes Affecting Authorization Server (+Client) Implementations

Following are specification changes in V1.0.1 that affect authorization servers, and possibly clients that interact with them as well.

AS Now Has Unique Redirect URI Endpoint for Claims Gathering (+Client)

Previously, the client was instructed to present the ordinary OAuth redirect_uri endpoint to which the AS should redirect requesting parties back after claims gathering, but this was ambiguously specified and incorrect. Now the client has a unique endpoint, `claims_redirect_uri`, that it needs to register. (144)

Permission Ticket Lifecycle Management (+Client)

Previously, little guidance was offered on how to manage permission tickets. Now some implications are explored, particularly as they relate to client interaction. (172) (Core Sec 3.2.2)

Requested Permission and Permission Ticket Matching

Previously, the matching of the "extents of access" of the requested permission registered by the RS and the permission ticket issued by the AS was implicit. Now it is spelled out. (175) (Core Sec 3.2)

Permission Ticket on Redirect Back to Client (+Client)

Previously, the AS was required to repeat the client's permission ticket back to it in a `ticket` property when offering a `redirect_user` hint in `error_details`. Now this is optional and the client is encouraged to ignore the property's value, preparatory to removing the property entirely in a future UMA version. The reason is that the value can't be guaranteed good; repeating the value was in order to save the client work; and having the client check the value would ultimately have caused both sides work for no gain. (205) (Core Sec 3.5.4.2)

PUT Means Complete Replacement

Previously, the requirement for an Update method in resource set registration to completely replace the previous resource set description was implicit. Now it is spelled out. (177) (RSR Sec 2.2.3)

Default-Deny for Authorization Data Issuance

Previously, a naive implementation could have resulted in accidental default-permit authorization data issuance in some cases. Now a default-deny authorization assessment model has been made explicit, with an example given of how implementations could get into trouble. (194) (Core Sec 3.5.2)

base64url-Encoded Claims (+Client)

Previously, the wording about base64url-encoding pushed claims was ambiguous about whether double-encoding was necessary in the case of claim formats that were already base64url-encoded. Now it has been clarified that double-encoding should not be performed. (206) (Core Sec 3.6.2)

Enhanced Security Considerations

Previously, the security considerations around accepting policy-setting context information from an incompletely trusted RS only covered "bad icon URIs". Now they cover all such policy-setting context information, following roughly the OAuth example. (151) (RSR Sec 4)

Previously, the security considerations around client-pushed claims were explored only in a very cursory fashion in the body of the text. Now they are treated at length in a new subsection. (160) (Core Sec 7.4.1)

Enhanced Privacy Considerations

Previously, little was said about privacy implications of requesting party claims being transmitted to the AS. Now this section has been greatly expanded. (211) (Core Sec 8.2)

Changes Affecting Resource Server (+Client) Implementations

Following are specification changes in V1.0.1 that affect resource servers, and possibly clients that interact with them as well.

Caveat About Resource Server API Constraint

Previously, the specification was missing an important caveat: Based on a client's initial RPT-free resource request, the RS needs to know the correct AS, PAT, and resource set ID to include in its follow-on call to the permission request endpoint at the AS. Thus, the API of the RS needs be structured so that it can derive this information from the client's request. Now this caveat appears in several locations. (161, 162, 225)

Adjustment of Other Resource Server API Constraints (+Client)

Previously, the specification wording was inconsistent and problematic regarding how the RS responds to a client request accompanied by no RPT or an RPT with insufficient authorization data (assuming permission request success). Now the ability not to respond at all is more fully acknowledged; all responses intended to be interpreted in an UMA fashion are required to be accompanied by a `WWW-Authenticate: UMA` header; the permission ticket is required to be returned in a new `ticket` parameter in that header; complete freedom is given regarding the RS's choice of HTTP status code; and only in the case of a 403 choice is a ticket in a JSON-encoded body suggested, preparatory to removing the body option in a future UMA version. The rationale for this somewhat dramatic set of changes is that the original prescription to return HTTP status code 403 was incorrect; the specification gave too little guidance about responses other than 403 responses to be useful for client interoperability; and its requirement to return the permission ticket in a JSON-encoded body regardless of expected content type was an issue. (163, 164, 168) (Core Sec 3.3.1)

Solution for Permission Registration Failure (+Client)

Previously, the specification gave no guidance on how the RS should respond to the client in case of permission registration failure at the AS. Now, if the RS responds at all, it is required to substitute a `Warning: 199 - "UMA Authorization Server Unreachable"` header for `WWW-Authenticate: UMA`. (176) (Core Sec 3.3.2)

Authorization Server URI to Return to Client (+Client)

Previously, the value of the `as_uri` property that the RS returns to the client was described somewhat vaguely as the authorization server's URI. Now it has been clarified to be the `issuer` URI as it appears in the AS configuration data of the AS. (199) (Core Sec 3.3.1)

New Security Considerations

Previously, the security considerations around accepting policy-setting context information from an incompletely trusted AS were not covered. Now they cover the `user_access_policy_uri` property, which is the only policy-setting context information passed from AS to RS. (185) (RSR Sec 4)

Specification Reorganizations

The specifications, particularly Core Sec 3, were reorganized in the fashion of OpenID Connect, with the goal of giving a subsection to every request and response message. Other notable changes include:

- Several “commentary” subsections were added, such as Core Sec 3.2.2 discussing permission ticket creation and management, and RSR Sec 2.1.2 discussing scope interpretation.
- A new section, Core Sec 9.2, registers the permissions property in the new OAuth token introspection IANA registry (this is in addition to its registration in the JWT claims registry).
- Core Sec 7.4.1 breaks out the new, more extensive security considerations discussion of pushed claims.
- Core Sec 8 now has subsections to make privacy considerations easier to find and understand.

Sections are presented in original V1.0 (black) Table of Contents order, mapped to their corresponding draft V1.0.1 sections (green). Where a V1.0.1 section or block of sections is repeated, it redistributes material previously appearing in the V1.0 sections under which the mentions appear.

Core Specification Reorganization

Found in Core V1.0 (go)
Find in Core draft V1.0.1 (go)

1. Introduction (go)
1.1. Notational Conventions
1.2. Terminology
1.3. Achieving Distributed Access Control
1.3.1. Protection API
1.3.2. Authorization API
1.3.3. Protected Resource Interface
1.3.4. Time-to-Live Considerations
1.4. Authorization Server Configuration Data
1. Introduction (go)
1.1 Notational Conventions
1.2 Terminology
1.3 Achieving Distributed Access Control
1.3.1 Protection API and Protection API Token
1.3.2 Authorization API and Authorization API Token
1.3.3 Protected Resource Interface and Requesting Party Token
1.3.4 Time-to-Live Considerations
1.4 Authorization Server Configuration Data

2. Protecting a Resource (go)
2. Protecting a Resource (go)

3. Getting Authorization and Accessing a Resource (go)
3.1 Client Attempts Access to Protected Resource

3. Getting Authorization and Accessing a Resource (go)
3.1 Client Attempts Access to Protected Resource

3.1.1. Client Request to Resource Server With No RPT (go)
3.1.1 Client Request to Resource Server With No RPT (go)
3.3 Resource Server Responds to Client (go)
3.3.1 Resource Server Response to Client on Permission Registration Success
3.3.2 Resource Server Response to Client on Permission Registration Failure

3.1.2. Client Presents RPT (go)
3.1.2 Client Request to Resource Server With RPT (go)
3.3 Resource Server Responds to Client (go)
3.3.1 Resource Server Response to Client on Permission Registration Success
3.3.2 Resource Server Response to Client on Permission Registration Failure
3.3.3 Resource Server Response to Client on Sufficiency of Authorization

3.2. Resource Server Registers Requested Permission With Authorization Server (go)
3.2 Resource Server Registers Requested Permission With Authorization Server (go)
3.2.1 Resource Server Request to Permission Registration Endpoint
3.2.2 Permission Ticket Creation and Management
3.2.3 Authorization Server Response to Resource Server on Permission Registration Success
3.2.4 Authorization Server Response to Resource Server on Permission Registration Failure

3.3. Resource Server Determines RPT's Status (go)
3.3.1. Token Introspection
3.3.2. RPT Profile: Bearer
3.4 Resource Server Determines RPT Status (go)
3.4.1 Token Introspection Process
3.4.2 RPT Profile: Bearer

3.4. Client Seeks Authorization for Access (go)
3.5 Client Seeks Authorization for Access (go)

3.4.1. Client Requests Authorization Data (go)
3.5.1 Client Request to Authorization Server for Authorization Data (go)
3.5.2 Authorization Assessment Process
3.5.3 Authorization Server Response to Client on Authorization Success
3.5.4 Authorization Server Response to Client on Authorization Failure

3.4.1.1. Authentication Context Flows (go)
3.6 Client Responds to Authorization Server's Request for Additional Information (go)
3.6.1 Client Redirects Requesting Party to Authorization Server for Authentication

3.4.1.2. Claims-Gathering Flows (go)
3.6 Client Responds to Authorization Server's Request for Additional Information (go)
3.6.2 Client Pushes Claim Tokens to Authorization Server (go)
3.6.3 Client Redirects Requesting Party to Authorization Server for Claims-Gathering

4. Error Messages (go)
4.1. OAuth Error Responses
4.2. UMA Error Responses
4. Error Messages (go)
4.1 OAuth Error Responses
4.2 UMA Error Responses

5. Profiles for API Extensibility (go)
5.1. Protection API Extensibility Profile
5.2. Authorization API Extensibility Profile
5.3. Resource Interface Extensibility Profile
5. Profiles for API Extensibility (go)
5.1 Protection API Extensibility Profile
5.2 Authorization API Extensibility Profile
5.3 Resource Interface Extensibility Profile

6. Specifying Additional Profiles (go)
6.1. Specifying Profiles of UMA
6.2. Specifying RPT Profiles
6.3. Specifying Claim Token Format Profiles
6. Specifying Additional Profiles (go)
6.1 Specifying Profiles of UMA
6.2 Specifying RPT Profiles
6.3 Specifying Claim Token Format Profiles

7. Compatibility Notes (go)
n/a

8. Security Considerations (go)
7. Security Considerations (go)

8.1. Redirection and Impersonation Threats (go)
7.1 Requesting Party Redirection and Impersonation Threats (go)

8.2. Client Authentication (go)
7.2 Client Authentication (go)

8.3. JSON Usage (go)
7.3 JSON Usage (go)

8.4. Profiles, Binding Obligations, and Trust Establishment (go)
7.4 Profiles and Trust Establishment (go)

n/a
7.4.1 Requirements for Trust When Clients Push Claim Tokens (go)

9. Privacy Considerations (go)
8. Privacy Considerations (go)
8.1 Resource Set Information at the Authorization Server
8.2 Requesting Party Information at the Authorization Server
8.3 Profiles and Trust Establishment

10. IANA Considerations (go)
9. IANA Considerations (go)

10.1. JSON Web Token Claims Registration (go)
10.1.1. Registry Contents
9.1 JSON Web Token Claims Registration (go)
9.1.1 Registry Contents

n/a
9.2 OAuth Token Introspection Response Registration (go)
9.2.1 Registry Contents

10.2. Well-Known URI Registration (go)
10.2.1. Registry Contents
9.3 Well-Known URI Registration (go)
9.3.1 Registry Contents

11. Acknowledgments (go)
10. Acknowledgments (go)

12. References (go)
12.1. Normative References
12.2. Informative References
11. References (go)
11.1 Normative References
11.2 Informative References

RSR Specification Reorganization

Found in RSR V1.0 (go)
Find in RSR draft V1.0.1 (go)

1. Introduction (go)
1.1. Notational Conventions
1.2. Terminology
1.3. Authorization Server Configuration Data
1. Introduction (go)
1.1 Notational Conventions
1.2 Terminology
1.3 Authorization Server Configuration Data

2. Resource Set Registration (go)
2. Resource Set Registration (go)

2.1. Scope Descriptions (go)
2.1.1 Scope Descriptions (go)

n/a
2.1.2 Scope Interpretation (go)

2.2. Resource Set Descriptions (go)
2.1 Resource Set Descriptions (go)

2.3. Resource Set Registration API (go)
2.3.1. Create Resource Set Description
2.3.2. Read Resource Set Description
2.3.3. Update Resource Set Description
2.3.4. Delete Resource Set Description
2.3.5. List Resource Set Descriptions
2.2 Resource Set Registration API (go)
2.2.1 Create Resource Set Description
2.2.2 Read Resource Set Description
2.2.3 Update Resource Set Description
2.2.4 Delete Resource Set Description
2.2.5 List Resource Set Descriptions

3. Error Messages (go)
4. Security Considerations
5. Privacy Considerations
6. IANA Considerations
7. Example of Registering Resource Sets
8. Acknowledgments
9. References
9.1. Normative References
9.2. Informative References
3. Error Messages (go)
4. Security Considerations
5. Privacy Considerations
6. IANA Considerations
7. Example of Registering Resource Sets
8. Acknowledgments
9. References
9.1 Normative References
9.2 Informative References

Pre-V1.0 Changes

Following is a catalog of notable changes to the specifications in the pre-V1.0 timeframe.

Core Changes

Internet-Draft Rev 11 to Rev 12

From I-D rev 11 to rev 12:

- Notable changes:
  - Enhanced the Security Considerations section.

Internet-Draft Rev 10 to Rev 11

From I-D rev 10 to rev 11:

- Breaking changes:
  - Section 3.4: not_authorized_permission error code: Changed to not_authorized.
  - RPT handling: Changed extensively to remove the RPT issuance endpoint and enable the authorization data request endpoint to do all RPT issuance duties. Permission ticket issuance is now handled on an "eager" basis, when a client either without an RPT or with an invalid or insufficient-authorization-data RPT approaches the RS seeking access. This affects several sections:
    - Section 1.4: configuration data
    - Section 3: introduction
    - Section 3.1.1 and 3.1.2: client approaching RS
    - Section 3.2: RS registering permission
    - Section 3.4: RPT issuance and authorization data addition
    - Section 5.2: Extensibility profile implications
  - Section 1.4:
    - Changed the claim_profiles_supported property in the configuration data to claim_token_profiles_supported
    - Changed the user_endpoint property in the configuration data to authorization_endpoint, to match the final IETF RFC 6749 name in OAuth 2.0
    - Changed the authorization_request_endpoint property in the configuration data to rpt_endpoint, to distinguish it more fully from the OAuth endpoint and to shorten it
    - (Also affects Section 5) Changed how uma_profiles_supported works, so that the API extensibility profiles don't have reserved keywords but rather use the regular URI mechanism for indicating profiles
  - Section 3.3.2:
    - Names of several properties in the permissions structure for the RPT "Bearer" token introspection response have changed to align them with JWT names: expires_at to exp, issued_by to iat
    - The JWT "scope" property at the top level is now disallowed in favor of "scopes" at the permissions level.
  - PAT and AAT OAuth scopes:
    - Renamed from URIs to simple strings: "uma_protection" and "uma_authorization"; the JSON scope description documents provided to enable the old URIs to resolve no longer have any relation to the UMA Core spec
- Other changes of note:
  - Section 3.1.1 and Section 3.1.2: Extraneous host_id removed from example of RS's response to client.
  - Enabled explicit use of OAuth-based authentication protocols such as OpenID Connect for OAuth protection driving PAT and AAT issuance.
  - Identifiers for spec-defined profiles now use https instead of http
  - Migrated the claim profiling spec's requesting party claims endpoint configuration data to the core spec, and made it optional to supply.
  - Migrated the claim profiling spec's "need_claims" extensions to the core spec, broadened it to "need_info", and gave it "error_details" hints in the core spec.
  - Section 3.1.1: Requirement for RS to return 403 to a tokenless client has been softened to a SHOULD.
  - Section 3.3.2: The token introspection response has been aligned with the latest token introspection spec. nbf has been added at the permissions level, exp is now optional, and all permissions-level properties that duplicate JWT-level claims in intent now get overridden by any JWT-level claims present in the response. Finally, the "permissions" JWT claim has been registered with IANA.
  - Extensive new redirect-pattern claims gathering support added
  - Extensive new security and privacy considerations added
  - Section 1.4:
    - issuer property: Now required to match the actual published location of the config data.
    - Dynamic client configuration: When OIDC dynamic client configuration is used, this is now more intelligently handled through a reserved keyword "openid" that indicates that the OIDC configuration data should be consulted for the relevant endpoint.
    - pat_grant_types_supported and aat_grant_types_supported: Broadened to allow them to be strings even when not based on the OAuth grant type strings, similarly to token profiles.
    - issuer property: Now required to match the actual published location of the config data.
    - Dynamic client configuration: When OIDC dynamic client configuration is used, this is now more intelligently handled through a reserved keyword "openid" that indicates that the OIDC configuration data should be consulted for the relevant endpoint.
    - pat_grant_types_supported and aat_grant_types_supported: Broadened to allow them to be strings even when not based on the OAuth grant type strings, similarly to token profiles.

Internet-Draft Rev 08 to Rev 09

From I-D rev 08 to 09:

- Breaking changes:
  - (Technically breaking but not expected to have huge impact:) TLS/HTTPS is now mandatory for the AS to implement in its protection and authorization APIs.
- Other changes of note:
  - It is no longer required for the client to redirect a human requesting party to the AS for the claims-gathering process.
  - A new claims profiling framework (now in a separate spec) describes how to leverage one of several common patterns for claims-gathering: client redirects the requesting party to AS, client pushes claims to the AS.

- A new framework for API extensibility, and a matching series of extensibility profiles, appears in the core spec. It enables tighter coupling between the AS and RS, AS and client, and RS and client, respectively, but only in a controlled manner to foster greater interoperability in such circumstances.
- The SHOULD for the usage of the SAML bearer token profile for PAT issuance is now just a MAY.
- In Section 4.2, the example was corrected to remove a wayward "status" : "error" property.
- Clarified that no request message body is expected when the client uses the RPT endpoint at the AS.
- Added a success example in Section 3.4.2 showing how authorization data is added and the RPT is simultaneously refreshed, a new capability.

Internet-Draft Rev 07 to Rev 08

From I-D rev 07 to 08:

- Breaking changes:
  - Section 1.3: TLS as defined and (mostly) required in OAuth (RFC 6749) is now a MUST in UMA for AS endpoints.

From I-D rev 06 to 07:

- Breaking changes:
  - Section 1.5: Some properties in the the authorization server configuration data have been renamed, and others broken out into multiple properties with different names. The wording around reserved keywords vs. URIs as string values was also cleaned up.
    - **oauth_token_profiles_supported**: broken out into two, **pat_profiles_supported** and **aat_profiles_supported.**
    - **uma_token_profiles_supported**: renamed to **rpt_profiles_supported.**
    - **oauth_grant_types_suppored**: broken out into two, **pat_grant_types_supported** and **aat_grant_types_supported.**
  - Section 3.4.2: Error code names were cleaned up.
    - **expired_requester_ticket**: renamed to **expired_ticket**.
    - **invalid_requester_ticket**: renamed to **invalid_ticket**.
  - Other changes of note:
    - Updated the token introspection spec citation and details.
    - Updated the OAuth threat model citation.
    - Enhanced the security considerations section.
    - Broaden from defining successful access as 200 OK to defining it as 2xx Success.
    - Explain that the PAT implicitly gives the "subject" of a requested permission.
    - Fix resource_set_registration_endpoint keyword mention. (It was missing the last work.)

Internet-Draft Rev 05 to Rev 06

From I-D rev 05 to 06:

- Breaking changes:
  - Section 1.5: The authorization server configuration data now allows for providing a dynamic client registration endpoint (now defined by the official OAuth dynamic client registration spec), rather than just serving as a flag for whether the generic feature is supported. The name changed to **dynamic_client_endpoint**.
  - Sections 3.1.1 and 3.1.2: The **am_uri** header has been renamed to **as_uri** due to terminology changes (see below).
  - Section 3.1.2: The OAuth error "insufficient_scope" is now a central part of the authorization server's response to a client with a valid RPT and insufficient scope. This aligns UMA more closely with OAuth as a profile thereof (stay tuned for more possible tweaks in this general area, e.g. in WWW-Authenticate).
- Other changes of note:
  - Terminology has been changed wholesale from UMA-specific terms to OAuth-generic terms.
    - Authorization manager (AM) is now authorization server.
    - Host is now resource server.
    - Authorizing user is now resource owner.
    - Requester is now client.
  - Some additional terms and concepts have been tweaked, enhanced and clarified.
    - Scope is now scope type (likely to change back due to feedback).
    - Authorization data is now defined as a generic category, of which permissions are an instance.
    - RPT now stands for requesting party token instead of requester permission token.
    - UMA is more explicitly defined as a profile of OAuth.
  - References have been added to the OAuth token introspection spec proposal, though it is not fully used yet (stay tuned for breaking changes here).
  - The resource set registration process (phase 1) has been moved to a separate modular spec that is designed to be usable by other OAuth-based technologies along with UMA.

RSR changes

Internet-Draft Rev 04 to Rev 05

From I-D rev 04 to rev 05:

- Breaking changes:
  - Changed the PUT method for the purpose of resource set creation at the authorization server, to POST. This had other rippling changes, such as removing the usage of If-Match, the precondition_failed error, ETag usage, and the privacy considerations warning about mapping real resource set names to obscured names that remove personally identifiable information.
  - Changed the name of the policy_uri property to user_access_policy_uri to differentiate it from the OAuth property of (formerly) the same name
- Other changes of note:

- Clarified that user_access_policy_uri is allowed on Create, Read, and Update, and also now allow it on Delete and List too.
- Enhanced the Security Considerations section.

Internet-Draft Rev 03 to Rev 04

From I-D rev 03 to rev 04:

- Breaking changes:
  - Removed the "status: xxx" property from all the AS responses in the RSR API.
- Other changes of note:
  - ("04" to "05") Added a new optional resource_uri parameter to the resource set description, to support resource discovery at an authorization server.
  - Scopes bound to resource set descriptions can now be strings rather than being required to be URIs that resolve to scope description documents.
  - The _rev property has been removed from the resource set registration API. It can be added back as an extension for those who want it.

Claim Profiles changes

Claim Profiles Rev 00

Claim Profiles 00:

- We decided not to progress this specification in its current form, so we will let it expire and will not reference it from Core.

Change History

Version	Date	Comment
Current Version (v. 39)	**Mar 11, 2018 22:39**	**Eve Maler**: Added the subsection "Only One Pushed Claim Token Now Allowed at a Time"
v. 38	Jan 09, 2018 22:16	**Eve Maler**: Added change history macro
v. 37	Jan 09, 2018 22:15	**Eve Maler**: Corrected header material; added links into specific Recommendation sections
v. 36	Nov 16, 2017 19:26	**Eve Maler**
v. 35	Nov 16, 2017 03:00	**Eve Maler**
v. 34	Nov 01, 2017 00:04	**Eve Maler**
v. 33	Oct 10, 2017 16:53	**Eve Maler**
v. 32	Oct 10, 2017 15:55	**Eve Maler**
v. 31	Sep 06, 2017 15:11	**Eve Maler**
v. 30	Aug 08, 2017 17:53	**Eve Maler**
v. 29	Aug 08, 2017 17:46	**Eve Maler**
v. 28	Aug 08, 2017 17:20	**Eve Maler**
v. 27	Jul 18, 2017 14:01	**Eve Maler**
v. 26	Jul 18, 2017 13:21	**Eve Maler**
v. 25	Jul 18, 2017 13:20	**Eve Maler**
v. 24	Jul 12, 2017 10:30	**Eve Maler**

v. 23	Jul 05, 2017 11:07	**Eve Maler**
v. 22	Jul 05, 2017 10:42	**Eve Maler**
v. 21	Jun 30, 2017 00:21	**Eve Maler**
v. 20	Jun 27, 2017 17:51	**Eve Maler**
v. 19	Jun 25, 2017 23:14	**Eve Maler**
v. 18	May 14, 2017 06:49	**Eve Maler**
v. 17	May 14, 2017 06:47	**Eve Maler**
v. 16	May 05, 2016 15:10	**Eve Maler**
v. 15	May 05, 2016 14:49	**Eve Maler**
v. 14	Jan 25, 2016 10:57	**Eve Maler**
v. 13	Sep 20, 2015 14:39	**Eve Maler**
v. 12	Sep 20, 2015 14:27	**Eve Maler**
v. 11	Sep 20, 2015 14:27	**Eve Maler**
v. 10	Sep 20, 2015 13:51	**Eve Maler**
v. 9	Sep 20, 2015 13:49	**Eve Maler**
v. 8	Sep 20, 2015 13:45	**Eve Maler**
v. 7	Sep 20, 2015 13:06	**Eve Maler**
v. 6	Sep 18, 2015 10:28	**Eve Maler**
v. 5	Sep 18, 2015 10:13	**Eve Maler**
v. 4	Sep 15, 2015 21:30	**Eve Maler**
v. 3	Sep 15, 2015 21:25	**Eve Maler**
v. 2	Sep 15, 2015 19:38	**Eve Maler**
v. 1	Sep 15, 2015 14:58	**Eve Maler**