Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 21 January 2026

C. Bunch
Emerald Groupware Project

20 July 2025

A Rich Authorization Request Framework for Groupware Services using OAuth 2.0
draft-bunch-groupware-scopes-01

Abstract

The OAuth 2.0 authorization framework is widely used to provide clients with delegated access to user data. However, the core specification leaves the definition of access scopes to individual service providers. This has led to a fragmented ecosystem for common groupware services (Mail, Calendaring, Contacts), where each provider uses proprietary, non-interoperable scope identifiers. Client applications, such as desktop mail clients, are forced to hardcode configurations for a small number of large providers, stifling innovation and harming open ecosystems.

This document proposes a framework that combines a minimal set of standardized OAuth 2.0 scopes with a detailed structure for use with OAuth 2.0 Rich Authorization Requests (RAR). This hybrid approach provides a simple baseline for legacy clients while enabling powerful, granular, and privacy-preserving authorization for modern clients. It defines RAR type values, actions, datatypes, and custom fields for specifying fine-grained permissions for groupware resources.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 21 January 2026.

Table of Contents

1. Introduction

While OAuth 2.0 [RFC6749] is ubiquitous, its reliance on provider-defined scopes has created a fragmented ecosystem for groupware. A client application cannot automatically configure itself for a new mail or calendar provider, as it has no standard way to request specific permissions like "read-only access to the 'Work' calendar."

This document specifies a solution that leverages the recent OAuth 2.0 Rich Authorization Requests (RAR) [RFC9396] standard to define structured, granular permissions. To maintain backward compatibility and provide a simple path for basic clients, this detailed RAR structure is paired with a minimal set of three high-level scopes.

This allows a client to request simple access (e.g., scope=mail) or to make a much more specific request, such as asking for permission to read and delete messages only from the "INBOX" folder and send email as a specific alias. This improves both security and user privacy by adhering to the principle of least privilege.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Groupware Scopes

This specification proposes three aggregate scope values to represent high-level access to a user's groupware data. These scopes are intended as a baseline for simple clients or as a companion to more detailed Rich Authorization Requests.

urn:ietf:params:oauth:scope:mail
: Requests access to the user's email. The specific level of access SHOULD be further specified using a Rich Authorization Request of type urn:ietf:params:oauth:auth-type:mail. If RAR is not used, this scope implies full access.

urn:ietf:params:oauth:scope:calendar
: Requests access to the user's calendar data. Granular permissions SHOULD be requested via a Rich Authorization Request of type urn:ietf:params:oauth:auth-type:calendar.

urn:ietf:params:oauth:scope:contacts
: Requests access to the user's contact data. Granular permissions SHOULD be requested via a Rich Authorization Request of type urn:ietf:params:oauth:auth-type:contacts.

3. Rich Authorization Request Types for Groupware

This document defines a set of type values and associated fields for use within the authorization_details array of an OAuth 2.0 Rich Authorization Request.

3.1. Custom Authorization Details Fields

This specification defines the following new fields for use within the authorization_details object to provide granular control over groupware resources.

collections (array of strings):
: This field specifies the names or unique identifiers of the collections (e.g., mail folders, calendars) to which the request is scoped. The special value "*" indicates all collections of the specified datatypes. To solve the initial client bootstrapping problem, the special string value "__default" (two underscores) may be used. If provided, the Authorization Server SHOULD grant access to the user's primary or default collection for the given type (e.g., the INBOX for mail, the default calendar for CalDAV). If this field is omitted entirely, access SHOULD be treated as if no collections were granted. To prevent ambiguity, Resource Servers SHOULD NOT allow users to create collections with names that begin with two underscores.

type_specific_details (object):
: This object contains parameters that are unique to a specific authorization type and do not fit into the other standard fields.

3.2. The 'mail' Authorization Type

This type is used to request granular access to a user's email.

type:
: urn:ietf:params:oauth:auth-type:mail

Associated Scope:
: urn:ietf:params:oauth:scope:mail

Defined datatypes:
: folder, message, flag

Defined actions:
list, create, read, update, delete

Contents of type_specific_details:
send_as_aliases (array of strings):
Specifies the email addresses the client is permitted to use in the From: header when performing a send action. The value "*" indicates any of the user's available sending addresses.

3.3. The 'calendar' Authorization Type

This type is used to request granular access to a user's calendars.

type:
urn:ietf:params:oauth:auth-type:calendar

Associated Scope:
urn:ietf:params:oauth:scope:calendar

Defined datatypes:
calendar, event, freebusy

Defined actions:
list, create, read, update, delete

3.4. The 'contacts' Authorization Type

This type is used to request granular access to a user's address books.

type:
urn:ietf:params:oauth:auth-type:contacts

Associated Scope:
urn:ietf:params:oauth:scope:contacts

Defined datatypes:
addressbook, contact

Defined actions:
list, create, read, update, delete

4. Security Considerations

The principle of least privilege SHOULD be followed. A client application SHOULD use Rich Authorization Requests to request the narrowest set of permissions required for their functionality. For example, a scheduling assistant application that only needs to find open time slots should request only the freebusy datatype and the list and read actions for the calendar type, limited to specific calendars if possible.

Authorization servers MUST ensure that a user has explicitly consented to the scopes and rich authorization details requested by a client. Resource Servers are responsible for parsing the access token and correctly enforcing the authorized permissions.

5. IANA Considerations

This document requests the registration of new values in several IANA-managed registries.

5.1. OAuth Scope Registry

This document requests registration of the following three values in the "OAuth Scope Registry".

5.1.1. urn:ietf:params:oauth:scope:mail

Name:
urn:ietf:params:oauth:scope:mail

Description:
Requests access to a user’s email.

Change Controller:
IETF

Reference:
This document.

5.1.2. urn:ietf:params:oauth:scope:calendar

Name:
urn:ietf:params:oauth:scope:calendar

Description:
Requests access to a user’s calendar data.

Change Controller:
IETF

Reference:
This document.

5.1.3. urn:ietf:params:oauth:scope:contacts

Name:
urn:ietf:params:oauth:scope:contacts

Description:
Requests access to a user’s contact data.

Change Controller:
IETF

Reference:
This document.

5.2. OAuth Authorization Details Types Registry

This document requests registration of the following three values in the "OAuth Authorization Details Types Registry".

5.2.1. urn:ietf:params:oauth:auth-type:mail

Name:
urn:ietf:params:oauth:auth-type:mail

Description:
Authorization details for accessing email resources.

Change Controller:
IETF

Reference:
This document.

5.2.2. urn:ietf:params:oauth:auth-type:calendar

Name:
urn:ietf:params:oauth:auth-type:calendar

Description:
Authorization details for accessing calendar resources.

Change Controller:
IETF

Reference:
This document.

5.2.3. urn:ietf:params:oauth:auth-type:contacts

Name:
urn:ietf:params:oauth:auth-type:contacts

Description:
Authorization details for accessing contact resources.

Change Controller:
IETF

Reference:
This document.

6. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/info/rfc6749>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9396] Lodderstedt, T., Richer, J., and B. Campbell, "OAuth 2.0 Rich Authorization Requests", RFC 9396, DOI 10.17487/RFC9396, May 2023, <https://www.rfc-editor.org/info/rfc9396>.

Appendix A. JSON Schema for Authorization Details

This appendix provides non-normative JSON Schema definitions for the authorization_details objects defined in this document.

A.1. Mail Authorization Details Schema

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "Mail Authorization Details",
  "type": "object",
  "properties": {
    "type": {
      "const": "urn:ietf:params:oauth:auth-type:mail"
    },
    "locations": {
      "type": "array",
      "items": {
        "type": "string",
        "format": "uri"
      }
    },
    "actions": {
      "type": "array",
      "items": {
        "type": "string",
        "enum": [
          "list",
          "create",
          "read",
          "update",
          "delete"
        ]
      }
    },
    "datatypes": {
      "type": "array",
      "items": {
        "type": "string",
        "enum": [
          "folder",
          "message",
          "flag"
        ]
      }
    },
    "collections": {
      "type": "array",
      "items": {
        "type": "string"
      }
    },
    "type_specific_details": {
      "type": "object",
      "properties": {
```

```
          "send_as_aliases": {
            "type": "array",
            "items": {
              "type": "string"
            }
          }
        }
      }
    },
    "required": [
      "type",
      "actions",
      "datatypes"
    ]
  }
```

A.2. Calendar Authorization Details Schema

```
  {
    "$schema": "https://json-schema.org/draft/2020-12/schema",
    "title": "Calendar Authorization Details",
    "type": "object",
    "properties": {
      "type": {
        "const": "urn:ietf:params:oauth:auth-type:calendar"
      },
      "locations": {
        "type": "array",
        "items": {
          "type": "string",
          "format": "uri"
        }
      },
      "actions": {
        "type": "array",
        "items": {
          "type": "string",
          "enum": [
            "list",
            "create",
            "read",
            "update",
            "delete"
          ]
        }
      },
      "datatypes": {
        "type": "array",
```

```
        "items": {
          "type": "string",
          "enum": [
            "calendar",
            "event",
            "freebusy"
          ]
        }
      },
      "collections": {
        "type": "array",
        "items": {
          "type": "string"
        }
      }
    },
    "required": [
      "type",
      "actions",
      "datatypes"
    ]
  }
```

A.3. Contacts Authorization Details Schema

```
  {
    "$schema": "https://json-schema.org/draft/2020-12/schema",
    "title": "Contacts Authorization Details",
    "type": "object",
    "properties": {
      "type": {
        "const": "urn:ietf:params:oauth:auth-type:contacts"
      },
      "locations": {
        "type": "array",
        "items": {
          "type": "string",
          "format": "uri"
        }
      },
      "actions": {
        "type": "array",
        "items": {
          "type": "string",
          "enum": [
            "list",
            "create",
            "read",
```

```
              "update",
              "delete"
            ]
          }
        },
        "datatypes": {
          "type": "array",
          "items": {
            "type": "string",
            "enum": [
              "addressbook",
              "contact"
            ]
          }
        },
        "collections": {
          "type": "array",
          "items": {
            "type": "string"
          }
        }
      },
      "required": [
        "type",
        "actions",
        "datatypes"
      ]
    }
```

Appendix B. Changes from Version -00

This version represents a significant revision based on initial community feedback. The core focus has shifted from defining a large set of granular scopes to a more flexible and modern framework that combines a minimal set of aggregate scopes with the structured data model of OAuth 2.0 Rich Authorization Requests (RAR) [RFC9396].

Major changes include:

* The document title has been updated to reflect the new focus on Rich Authorization Requests.

* The number of proposed scopes has been reduced to three primary aggregate scopes (mail, calendar, contacts) to provide a simple baseline for all clients.

* A new major section has been added defining RAR type values, actions, datatypes, and custom fields (collections, type_specific_details) that allow for fine-grained permission requests.

* The specification is now more closely aligned with the structure and intent of RFC 9396, including the use of URIs in the locations field.

* An appendix has been added with non-normative JSON Schema definitions for the proposed authorization_details objects.

Author’s Address

Clinton Bunch
Emerald Groupware Project
Email: cdbunch@emeraldgroupware.org