OAuth Working Group
Internet-Draft
Intended status: Standards Track
Expires: 27 June 2025

H. HM

24 December 2024

OAuth 2.0 Extension for AI Model Access
draft-hemanth-oauth-ai-scopes-00

Abstract

This document defines an extension to OAuth 2.0 for delegating scoped access to AI model APIs. It introduces a standardized scope syntax, resource indicators for AI providers, and token constraints suitable for AI workloads including spend limits and model restrictions.

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 27 June 2025.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

The proliferation of AI model APIs (OpenAI, Anthropic, Google, Mistral, etc.) has created a need for secure delegation of API access. Current approaches involve sharing API keys directly with applications, which:

* Exposes master credentials to third parties

* Provides no usage limits or audit trail

* Cannot be scoped to specific models or capabilities

* Cannot be revoked without rotating the master key

This specification extends OAuth 2.0 to address these concerns by defining:

1. A standard scope syntax for AI model access

2. Resource indicators for AI providers

3. Token metadata for usage limits and spending caps

4. Security considerations specific to AI workloads

1.1. Terminology

AI Provider A service offering AI model APIs (e.g., OpenAI, Anthropic)

Model A specific AI model (e.g., gpt-4, claude-3, gemini-pro)

Capability A function offered by a model (chat, embeddings, images, audio)

Master Key The user's API key for a provider

Delegated Token An OAuth access token with AI-specific scopes

1.2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Scope Syntax

2.1. AI Scope Format

AI-specific scopes follow this syntax:

ai:<provider>:<model>:<capability>

Examples:

* ai:openai:gpt-4:chat - Chat completions with GPT-4

* ai:anthropic:claude-3-opus:* - All capabilities for Claude 3 Opus

* ai:openai:*:embeddings - Embeddings with any OpenAI model

* ai:*:*:chat - Chat with any provider/model

2.2. Provider Identifiers

Provider	Identifier
OpenAI	openai
Anthropic	anthropic
Google AI	google
Mistral	mistral
Groq	groq
Together AI	together
Cohere	cohere

Table 1

2.3. Capability Identifiers

Capability	Description
chat	Chat/text completions
embeddings	Vector embeddings
images	Image generation
audio	Audio transcription/synthesis
vision	Multimodal/vision
code	Code generation

Table 2

3. Token Metadata

3.1. Token Introspection Response Extensions

The token introspection response ([RFC7662]) is extended with:

```
{
  "active": true,
  "scope": "ai:openai:gpt-4:chat",
  "ai_limits": {
    "monthly_spend_usd": 100.00,
    "daily_spend_usd": 10.00,
    "requests_per_minute": 60,
    "requests_per_day": 1000,
    "max_tokens_per_request": 4096
  },
  "ai_usage": {
    "spend_this_month_usd": 23.45,
    "spend_today_usd": 2.10,
    "requests_this_minute": 3,
    "requests_today": 156
  }
}
```

3.2. Limit Fields

Field	Type	Description
monthly_spend_usd	number	Maximum spend per calendar month
daily_spend_usd	number	Maximum spend per day
requests_per_minute	integer	Rate limit (RPM)
requests_per_day	integer	Daily request limit
max_tokens_per_request	integer	Per-request token limit

Table 3

4. Authorization Request

4.1. Additional Parameters

Parameter	Type	Description
ai_limits	JSON	Requested limits (as defined in Section 3.2)
ai_reason	string	Human-readable reason for access

Table 4

Example authorization request:

```
GET /authorize?
  response_type=code&
  client_id=app123&
  scope=ai:openai:gpt-4:chat&
  ai_limits={"monthly_spend_usd":50}&
  ai_reason=Code+assistant+for+IDE
```

5. Resource Server Requirements

5.1. Proxy Architecture

The resource server (authorization server or dedicated proxy) MUST:

1. Validate the OAuth access token
2. Verify the requested operation matches token scopes
3. Check usage against token limits
4. Substitute the master API key
5. Proxy the request to the AI provider
6. Log usage for auditing
7. Update usage counters

5.2. Error Responses

When limits are exceeded:

```
{
  "error": "ai_limit_exceeded",
  "error_description": "Daily spend limit of $10.00 exceeded",
  "ai_usage": {
    "spend_today_usd": 10.23,
    "daily_spend_usd": 10.00
  }
}
```

6. Security Considerations

6.1. Token Binding

For high-security deployments, tokens SHOULD be sender-constrained using:

* DPoP ([RFC9449])

* mTLS ([RFC8705])

6.2. Prompt and Response Handling

Resource servers:

* MUST NOT log prompt or response content by default

* MUST encrypt any logged content at rest

* SHOULD provide configurable retention policies

* SHOULD support zero-logging mode

6.3. Master Key Protection

* Master keys MUST be encrypted at rest

* Master keys MUST NOT be exposed in logs or error messages

* Key rotation SHOULD be supported without token invalidation

7. IANA Considerations

7.1. OAuth Scope Registration

This specification registers the "ai" scope prefix in the OAuth Parameters registry.

7.2. AI Provider Registry

This specification requests the establishment of a registry for AI provider identifiers.

8. References

8.1. Normative References

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

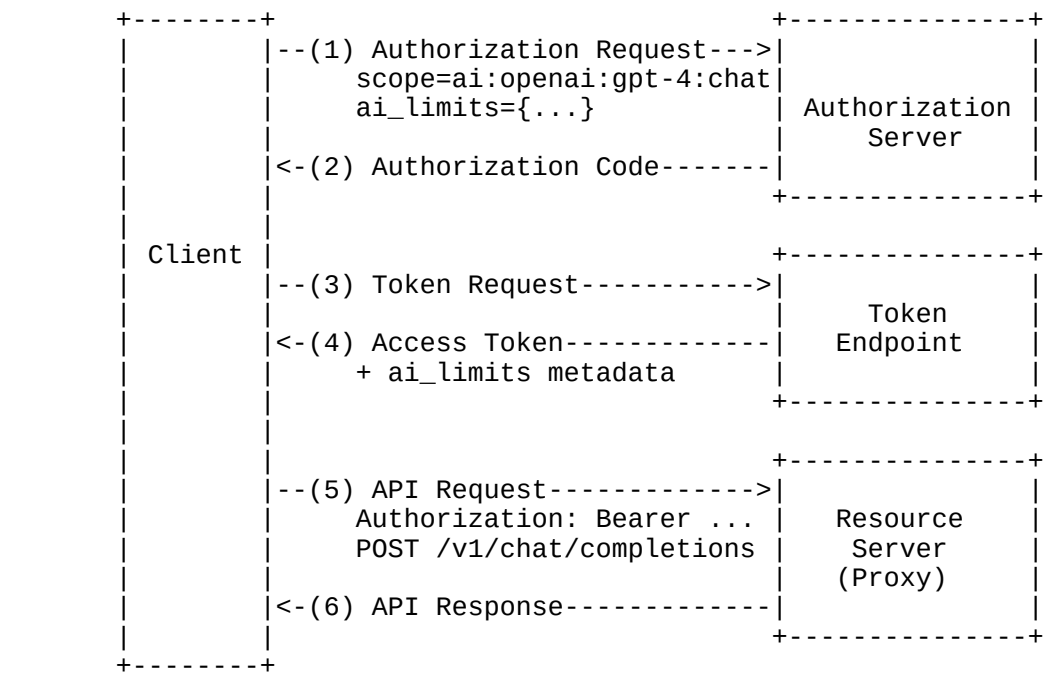
[RFC6749] Hardt, D., "The OAuth 2.0 Authorization Framework", RFC 6749, October 2012, <https://www.rfc-editor.org/info/rfc6749>.

[RFC7662] Richer, J., "OAuth 2.0 Token Introspection", RFC 7662, October 2015, <https://www.rfc-editor.org/info/rfc7662>.

[RFC8705] Campbell, B., Bradley, J., Sakimura, N., and T. Lodderstedt, "OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens", RFC 8705, February 2020, <https://www.rfc-editor.org/info/rfc8705>.

[RFC9449] Fett, D., Campbell, B., Bradley, J., Lodderstedt, T., Jones, M., and D. Waite, "OAuth 2.0 Demonstrating Proof of Possession (DPoP)", RFC 9449, September 2023, <https://www.rfc-editor.org/info/rfc9449>.

Appendix A. Example Flow



Acknowledgements

The author would like to thank the OAuth Working Group for their foundational work on authorization frameworks.

Author’s Address

Hemanth HM
Email: hemanth.hm@gmail.com
URI: https://h3manth.com