Gimel Foundation

Gimel Foundation gGmbH i.G.
GiFo-Request for Comments: 0111
Obsoletes: -
Category: Standards Track
ISBN: 978-3-00-084039-5

Dr. Goetz G. Wehberg
Digital Supply Institute
1. August 2025

The GAuth 1.0 Authorization Framework

Abstract

The GAuth authorization framework enables an artificial intelligence (AI, e.g., a digital agent, agentic AI or humanoid robot, respectively) to legitimize its power of attorney towards any other application including other AI and/or any other third party, including humans, on behalf of the owner of that AI by orchestrating an approval interaction, i.e. by owner allowing the AI to act and/or decide on its own behalf and legitimizing towards the relying third party, transparently and verifiably.

Status of This Memo

This is a Gimel Foundation Standards Track document.

This document is a product of the Gimel Foundation (GiFo). It represents the current consensus of the Gimel Foundation community. It has performed review and has been approved for publication.

Information about the status of this document, any errata, and how to provide feedback on it may be obtained at https://gimelfoundation.com or https://github.com/Gimel-Foundation.

Legal notice

Notational Conventions

The key words "Must", "Must Not", "Required", "Shall", "Shall Not", "Should", "Should Not", "Recommended", "May", and "Optional" in the following specification are to be interpreted as described in IETF` s RFC 2119.

Table of Contents

***

1. Scope

GAuth concerns the technical field of AI and particularly the governance of AI. With the increasing prevalence and performance of AI, there is a growing need for effective control and governance mechanisms to ensure their safe and responsible use. GAuth offers the solution for this. The GAuth standard offers an AI control protocol for digital agents, agentic teams of agents, humanoid robots and any other development of AI, which acts and decides on behalf of humans and/or organizations, which are being represented by humans. It acknowledges that the ultimate accountability of such actions and/or decisions are with such humans and/or organizations, respectively.

GAuth builds on the following standards as building blocks, thus is connected to these standards but adds distinguished complements in terms of its proprietary content, value-added, IP rights and overall license conditions. Building blocks include:

- OAuth or its alternatives, including but not limited to
  - RFC 6749
  - RFC 7636
  - Best Practices for OAuth 2.0 Security
- OpenID Connect or its alternatives, including but not limited to
  - OpenID Connect Discovery 1.0
  - OpenID Connect Dynamic Client Registration
  - OpenID Connect Session Management
- MCP or its alternatives, including but not limited to
  - MCP Implementation on Github (https://github.com/modelcontextprotocol)

2. Limitations on the right to make derivative works (Exclusions)

GAuth is freely available as an open-source solution based on Apache 2.0 with the following exclusions. Users of GAuth Must Not – whether directly or indirectly – integrate:

- The use of Web3, i.e., blockchain technology (including respective web3 tokens and smart contracts) for the extended token of GAuth,
- the use of AI or AI operators, i.e.,
  - AI that controls the entire process of an AI deployment life cycle,
  - AI that tracks actions and/or decisions regarding authorization compliance,
  - AI that assures the quality of the outcome of AI engagements and
- DNA-based identities or identities based on genetic data, respectively, which offer particularly secure biometrics
  - incl. AI that tracks the quality of DNA based identities
  - incl. AI that tracks risks in terms of identity theft or any other risks

These exclusions are excluded from the GAuth open source standard. Users Must Not use, integrate of add all or some of these exclusions in any form to the GAuth standard without a separate licensed permission in writing. Users Must exclude these exclusions from any implementation of GAuth unless licensed separately. Gimel Technologies GmbH offers separate solutions for this purpose, e.g. in terms of Gimel ID, G-Agent and GAuth+. These separate solutions are being protected by copyright and patent law.

The use of Gimel ID, specifically, provides a next level of assurance for identity verification, which can, e.g., add to or build on the Level of Assurance 4 of the Authentication Context Class Reference by OpenID Connect (ACR_LOA_4) in terms of a new level 5 (i.e., ACR_LOA_5).

In this context it is worth mentioning that any agentic AI to be authorized can comprise a lead AI agent and at least one team AI agent. A “lead AI agent” refers to a higher-level AI component within a hierarchical agent architecture that acts as a central coordination instance. The lead AI agent can make strategic decisions, orchestrate the AI system and consider complex dependencies between different automated actions. Authorization in this environment basically can be carried out centrally by GAuth for all AI units, or in a decentral mode, i.e. by the lead agent or subsequent cascaded hierarchy levels. The GAuth standard Must only be applied for centralized authorization, i.e., all AI units Must be authorized centrally by GAuth. The transfer of authorization authority to the AI-team lead or any other component of the AI system, like decentralized AI units or AI team members, Must Not be deployed and is subject to the exclusions of the GAuth standard. This protects the central authorization instance of GAuth and ensures the independence of the GAuth protocol. Any AI-controlled GAuth protocol Must be licensed additionally under separate license conditions in line with the exclusions of this specification.

3. Nomenclature

The following paragraphs explain definitions for technical terms used by the GAuth standard. The definitions Should Not be understood as limiting the scope of application or technical variants, but rather as pointers to some ways of understanding implementations of GAuth without excluding interpretations that are not mentioned in the definitions. Accordingly, GAuth also includes other possible implementations than the variants mentioned in the following paragraphs.

As the GAuth protocol builds on the OAuth protocol, GAuth builds on the role definitions of OAuth and further develops it as follows (adjustments or additions in *italic*):

“Resource owner”: An entity capable of granting access to a protected resource, *entering a legally binding transaction and accepting a decision or an action or any other impact suggested by a client*. When the resource owner is a person, it *can be* referred to

as an end-user. *The resource owner is subject of an AI` s (requested) transaction, decision or action*.

“Resource server”: The server hosting the protected resources *or any other asset being impacted by client` s transactions, decisions or actions*, capable of accepting and responding to protected resource requests using access *or extended* tokens. *The resource server is object of AI` s (requested) transaction, decision or action*.

“Client”: An application *or AI (e.g., digital agents, agentic AI or robots)* making protected resource requests - *including requests to enter a transaction and accept actions or decisions taken* - on behalf of the resource owner and with its authorization. The term "client" does not imply any implementation characteristics (e.g., whether the application executes on a server, a desktop, or other devices).

“Authorization server”: The server issuing *extended* tokens to the client after successfully authenticating the resource owner *as well as client* and obtaining authorization.

The OAuth protocol flow is shown in Figure 1, which provides a kind of baseline for GAuth.

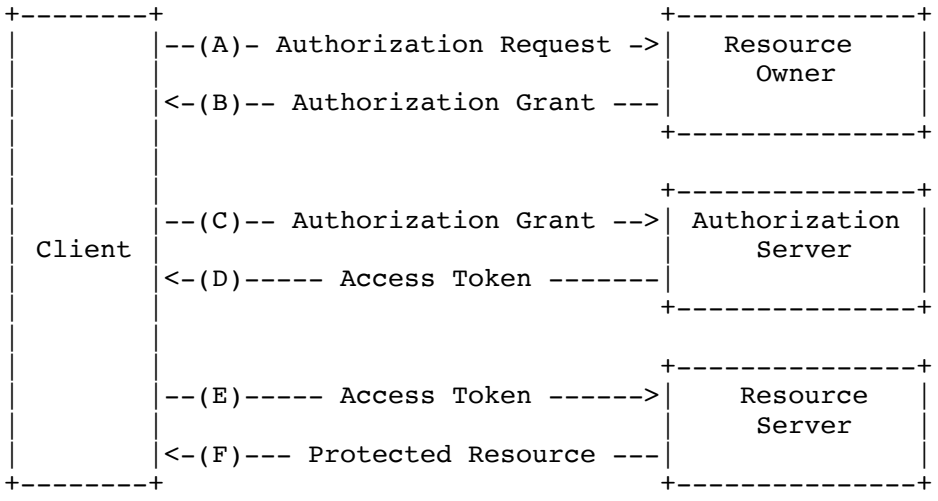

Figure 1: Abstract OAuth protocol flow (Source: RFC 6749, IETF / D. Hardt)

Moreover, GAuth defines “extended token” as credential used to serve a specific request. Extended tokens represent specific scopes and durations of authorization, granted by the resource owner, and enforced by the resource server and authorization server. As a digital representation in terms of set of data or any other form of representation an extended token summarizes the authorization for a specific request, potentially including access rights but beyond and more comprehensive. Technically, extended tokens May work like access tokens of OAuth, however, are not limited to it.

A “request” by a client is credentializing an application to enter a transaction, accept a decision or execute an action with the approval of the resource owner and the support of

the resource server, thus asking for commitment and/or permission, e.g., to sign, execute, run, produce, deliver, support, communicate, share, grant, etc. or anything else an AI or its principal and/or delegate can do. Technically, a request May work like requests of OAuth, however, is not limited to it.

An authorization “grant” is a credential representing the resource owner's authorization (to enter a transaction, accept a decision or support an action of the client or the client owner) used by the client to obtain an extended token. Technically, a grant May work like grants of OAuth, however, is not limited to it.

The protocol for issuing and managing ID tokens May work like OpenID Connect (e.g., Authorization Code Flow or Implicit Flow) or its alternatives (e.g., standards of uPort, DIF and its implementations based on Apache 2.0), however, is not limited to it.

In addition to this nomenclature, GAuth uses following specific roles:

The “client owner” defines the owner of the AI system that authorizes the AI system to enter transactions, act and take decisions in line with the authorization of the Client Owner.

The “owner`s authorizer” is the authorizer of the client owner or resource owner, respectively, and defines the power of attorney of the client owner or resource owner, e.g. its statutory authority.

Overall, the “P*P architecture” describes various abstract roles within the GAuth protocol and is referred to as “Power*Point” to emphasize the aspect of granting power of attorney, comprehensively (instead of using the wording “Policy*Point” regarding systems access rights):

- Power Enforcement Point (PEP) – usually the application, AI system or an intermediary that asks the PDP for a decision and enforces its result. GAuth differentiates between supply- and demand-side PEP. The client itself Must make sure it decides and acts in line with its authorization, thus enforces compliance from the supply-side. The resource owner and/or resource server Must check authorization compliance of the transactions, actions and decisions of the client and its owner as demand-side.
- Power Decision Point (PDP) – the authorization instance that grants authorization based on a series of inputs and makes decisions or grants approvals regarding the power of an AI. Typically, the PDP is the client owner. If the resource server is also an AI, the resource owner can be a PDP too.
- Power Information Point (PIP) – provider of data that contributes to the approval decision. Typically, the authorization server.
- Power Administration Point (PAP) – administrative level for the creation and management of authorization policies. Authorizing the client owner. Typically, the

PAP is the owner`s authorizer, i.e. the authorizer of the client owner and potentially also of the resource owner.

- Power Verification Point (PVP) – verification of the identities that perform a specific role along the GAuth processing. E.g., a trust service provider that also runs the authorization server.

These roles are compatible with current open source standards of OAuth, OpenID Connect, MCP and its alternatives, however, go beyond.

4. Why GAuth

AI like digital agents, agentic AI and humanoid robots can perform complex tasks autonomously, i.e., entering transactions, making decisions and performing actions. Humanoid robots represent a form of physical manifestation of digital agents. The capabilities of such AI poses challenges, particularly regarding control and accountability for the transactions, decisions and/or actions of these systems. AI governance aims to create frameworks and processes that ensure the ethical, safe, and lawful use of AI.

A central aspect of AI governance is the authorization and legitimization of AI. This involves clearly defining and documenting the granted powers, authority, and permitted scope of transactions, decisions or actions of an AI and on whose behalf it acts. This is particularly relevant in areas where AI acts on behalf of humans or organizations and makes potentially far-reaching decisions.

Existing approaches to AI governance focus mainly on establishing general principles and creating transparency. These solutions reach their limits when it comes to defining, processing, and monitoring the specific powers and scope of action of an AI in specific individual cases. The current Human-in-the-Loop approach is suggesting that AI is only supporting humans, with humans taking final decisions. This approach, however, limits the potential of AI to act autonomously. It comes with the risk that the accountable human gets used to rely on AI and to not question the outcome anymore. As much as AI acts autonomously without a proper governance, it can create risks of organizational fault and/or trust damages.

Current authorization protocols such as OAuth 2.0 (OAuth) offer access control options, but they are not specifically designed to meet the requirements of advanced AI and their governance. They primarily address the question of whether a system is allowed to access certain resources, but do not consider the more complex aspects of the decision-making powers and authority of independently acting AI. While OAuth typically integrates the OpenID Connect standard for verifying authorizers, the focus on system access remains.

In this context, the Model Context Protocol (MCP) was developed by the company Anthropic together with a developer community and represents an open standard that enables developers to establish bidirectional connections between data sources and AI-supported tools. Although it represents a step forward in the integration of AI, it does not comprehensively address governance aspects, in particular the question of authorizing and legitimizing AI for its decisions or actions. MCP applications typically use OAuth together with OpenID Connect or comparable standards.

Due to inadequate AI governance, both the combination of MCP, OAuth and OpenID Connect or comparable alternative standards are reaching their limits. It is not sufficient to limit AI authorization to access rights. Access rights are limited to answering the question “is this subject allowed to perform this action with this resource?”

5. What GAuth is

Autonomously acting AI evaluates, makes decisions, enters transactions and acts. Therefore, a comprehensive power-of-attorney mechanism Must cover these rights, i.e., answer the question "from whom has this AI received the power of attorney to make certain decisions or take certain actions (individual versus general power of attorney, registered office of the company, authorized representative/authorizing party, etc.), which decisions it is allowed to make and how, what kind of transactions it is permitted to enter and which actions it is allowed to perform with which kind of a specific resource, human or other agent (e.g., signing authority, authority to issue instructions, “need-to-do” or “do-unless” obligations)?”, not limited to it. This also raises the aspect of the “authority of the authorized representative or authorizing party,” i.e., a kind of second-level approval that ensures a dual control principle when using AI. A more comprehensive standard is therefore needed that contains the basic powers from which authorization can be derived in individual cases. This enables the relying party in terms of any subject or even object of an AI decision to exercise transparent control and verify the authorization of the client. Agents Must work within the limits and powers defined by the authorizing party (and, if applicable, their principal). Even if one agent authorizes another agent, a human being Must be at the top of such authorization cascade and thus ultimately be accountable. This is important to reduce the risks of organizational fault and avoid damage to trust.

GAuth integrates the specific aspects of comprehensively authorizing an AI, i.e., it takes all necessary elements and roles into account in an appropriate manner. In this respect, it complements the current governance framework. The verification of the identity of the authorizing parties, their secure authentication, transparent authorization of AI (beyond system access), and its legitimation (proof of authority by the AI to act compliantly) are closely related, as it is not sufficient to prove certain powers if the authorizing identity is

not clearly verified. The authorized AI Must be able to reliably prove the fact and scope of its authorization to act legitimately.

The GAuth protocol can be compared with the procedures of a commercial register for companies, which records the powers of a managing directors and authorized signatories. GAuth uses an authorization server to record the powers of action and decision-making of an AI. In this sense, GAuth represents a “commercial register for AI systems” that globally discloses the powers of attorney of AI, i.e. what a digital agent is supposed to sign, decide and do. It can be verified by any relying party having access to the authorization server, assuring the decisions or action of the respective AI has been authorized, thus behaving in compliance with its legitimized powers.

6. How GAuth works

GAuth is used to model comprehensive authorization concepts with their corresponding data structures. This requires careful design to capture the legal nuances. Policies, attributes, roles (P*P, etc.) and other criteria Should be used for the comprehensive mapping of power of attorney. For example, the delegation functions of GAuth can represent power of attorney relationships in which the principal (power of attorney grantor, i.e. client owner and/or owner` s authorizer) transfers certain powers to the agent (power of attorney recipient).

GAuth includes the following, not limited to it:

- Issuer, i.e. the individual or organization granting authority (i.e., owners or authorizers)
- Grantee as the AI system receiving authority (i.e., client and resource server)
- Successor as an optional attribute to name a backend AI if the primary AI (client or resource server) is unable to act
- Scope as to transactions, decisions or actions the AI is allowed to perform, including details geographic constraints or other conditions
- Delegation guidelines that specify principles associated with powers transferred
- Restrictions that define the limits of the transferred powers, e.g. value limits
- Validity period in terms of time restrictions for temporary powers of attorney
- Required attestations or witnesses, e.g. notary
- Version history of authorities transferred to track its history
- Revocation status which shows whether the power of attorney is still valid

A corresponding verification of the power of attorney by the relying party (resource owner / server or client owner or client, respectively) Must then consider, among other things, the following:

- Verification of powers – confirmation that the power of attorney is valid and active

- Verification of scope – ensuring that the requested action or decision taken falls within the scope of the powers transferred
- Status of the principal – verification of the principal's legal capacity and the position of the authorized representative
- Revocation handling – verification that the power of attorney has not been revoked

GAuth enforces the rules for powers of attorney mathematically and captures legal subtleties such as fiduciary duties, integrity requirements, or complex differences between jurisdictions.

The following description sets out the abstract GAuth protocol flow. This description is not intended to limit the scope of the GAuth standard, i.e., it also encompasses combinations and modifications of the abstract flow described herein. The GAuth protocol integrates all roles of the P*P architecture. GAuth comprises several consecutive steps, which are shown in logical order (Figure 2):

One-off steps to subscribe at authorization server (note: reference to selected building blocks of OpenID Connect in italic):

I. *Owner`s authorizer proves identity towards authorization server. Authorization server verifies.*
II. Owner`s authorizer proves authorization to authorization server. Authorization server verifies, e.g. via commercial register.
III. *Client owner proves identity towards authorization server. Authorization server verifies.*
IV. Client owner proves authorization to authorization server. Authorization server verifies, e.g. via owner`s authorizer.
V. Client owner authorizes client via authorization server, including sharing its identity and prompting of client.
VI. *Resource owner proves identity towards authorization server. Authorization server verifies.*
VII. Resource owner proves authorization to authorization server. Authorization server verifies, e.g. via owner`s authorizer.
VIII. Resource owner authorizes resource server via authorization server, including sharing its identity and prompting of resource server.

Request-specific steps to use authorization server (note: reference to selected building blocks of OAuth in italic):

a. *Client requests* specific *authorization from the resource owner*, in line with its general powers*. Even better than requesting authorization to the resource owner it can be requested to the resource server as intermediary.*

b. Resource owner or resource server, respectively, validates via authorization server the specific requests is compliant with the general powers of the client. Authorization server shares powers of clients, authorized by client owner.
c. *Client receiving an authorization grant from resource owner or server, which is a credential representing the resource owner's authorization.*
d. *Client requesting* an extended token *by authenticating with the authorization server and presenting the authorization grant.*
e. *Authorization server authenticates the client and validates the authorization grant, and if valid, issues* an extended token.
f. Client validates via authorization server the specific grant is compliant with the powers of the resource owner or resource server, respectively. Authorization server shares powers of resource owner or resource server, respectively, authorized by resource owner.
g. *Client requests* entering the transaction and/or contributing to respective decision or action from the resource server and *authenticates by presenting the* extended *token*.
h. *Resource server validates* the extended token, *and if valid, serves the request.*
i. Authorization server tracks compliance of client and/or resource server based on approval rules

Figure 2: Abstract GAuth protocol flow

While several embodiments have been described, it is understood that various modifications May be made for implementing it without departing from the spirit and scope of GAuth. Accordingly, alternative implementations also fall within the scope of GAuth.

7. Benefits

GAuth provides several benefits, which can be summarized by following adjectives:

Practical: GAuth offers several key advantages over the current state of the art. First, combining release rules stored on an authorization server with more comprehensive power-related approval rules or techniques enables relying parties to approve AI`s actions and decisions in a controlled manner. This represents a significant improvement over traditional governance approaches, which are often limited to generic, intransparent systems or general governance principles, thus not offering real practical help for the daily operations of an effective AI governance.

Comprehensive: GAuth addresses the limitations of current authorization protocols such as OAuth, which are focused on access control and do not sufficiently consider

the more complex aspects of AI`s decision-making powers. By combining server-based approval rules and learning mechanisms, GAuth creates a comprehensive basis for authorizing and legitimizing AI that goes far beyond simple access control mechanisms.

Verifiable: GAuth ensures a high degree of transparency towards relying parties, and an independent management of approval rules. This directly addresses the challenges of existing AI governance solutions, which often struggle to define and monitor specific powers and authorities of AI in individual cases in a comprehensible manner. The enforcement of a compliant behaviour of the AI from both sides, supply and demand side, facilitates both a trustful delegation of authority as well as secure collaboration with autonomously acting AI together with relying parties.

Automated: Another significant advantage of GAuth is that the protocol, more specifically the authorization server, can learn from experience and continuously automate its decision-making, based on a proper set of rules (not limited to it, in line with GAuth's exclusions). This leads to significantly higher efficiency in the approval of automated actions than would be possible with today`s standards.

Compounding: GAuth builds on current standards like OAuth and OpenID Connect, so that it is a compounding development of existing authorization protocols and architectures, not "going back to square one". It leverages on the strengths of existing open-source solutions, complementing it rather than competing.

Upgradable: The - within this specification - out-scoped features of GAuth (exclusions) can be upgrading its open-source protocol and even increase security by using web3 technology, DNA-based identities as well as AI in the context of an independent orchestration of the protocol itself.

8. Next steps

Requests, grants as well as extended token attributes of GAuth and the methods used to comprehensively authorize are beyond the scope of this specification and are being defined by subsequent specifications.

New developments such as post-quantum cryptography (e.g., by the National Institute of Standards and Technology / NIST of the United States of America) and next-level AI models (e.g., based on the Joint Embedding Predictive Architecture / JEPA from Yann LeCun) are compatible with GAuth, yet to be considered with its implementations.

WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

* * *

ISBN