IETF 123 Madrid
July 2025

Aaron Parecki
Daniel Fett
Joseph Heenan

# OAuth Client ID Prefix

https://datatracker.ietf.org/doc/draft-parecki-oauth-client-id-prefix/
**draft -00 but really -02**


IETF 123 Madrid
July 2025

Presented by Brian Campbell,
who is present on-site

Aaron Parecki
Daniel Fett
Joseph Heenan

# OAuth Client ID Prefix

https://datatracker.ietf.org/doc/draft-parecki-oauth-client-id-prefix/
draft -00 but **really -02**

# Recap

Presented draft-parecki-oauth-client-id-metadata-document at IETF 121

And presented Client ID Scheme at an Interim meeting in January 2025 in response to the feedback from IETF 121

Presented at IETF 122

Renamed to Client ID Prefix after IETF 122

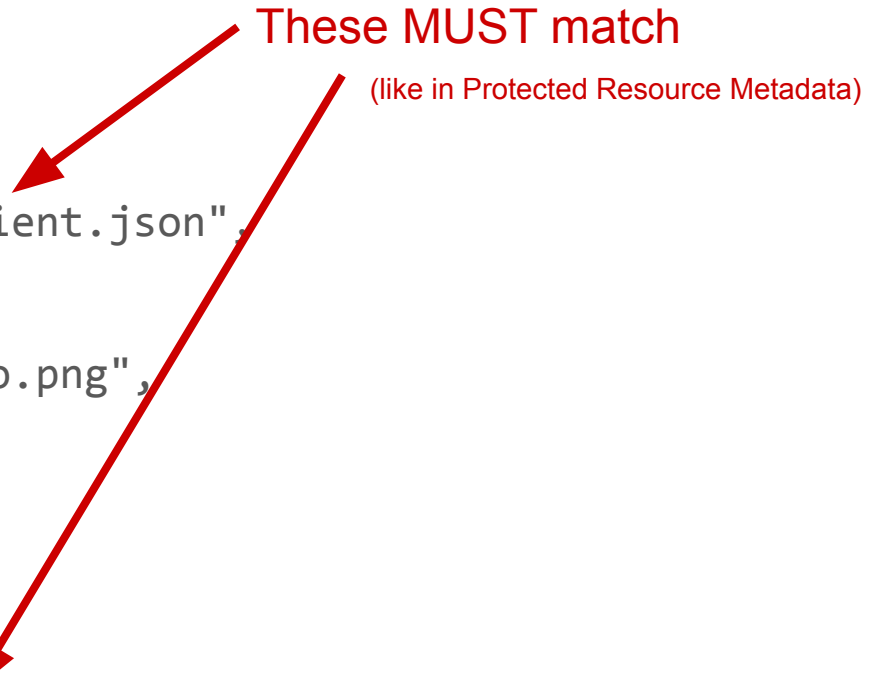
You are here at IETF 123

# Client ID Metadata Document

- The client publishes their metadata (Dynamic Client Registration vocabulary) at a URL
  - https://www.iana.org/assignments/oauth-parameters/oauth-parameters.xhtml#client-metadata
  - Doesn't have to be at a predefined path, but it should be a "stable URL", and may be displayed to the end user
- This URL is used as the Client ID in the authorization request
  - `…&client_id=https://example-app.com/client.json&scope=…`

https://datatracker.ietf.org/doc/draft-parecki-oauth-client-id-metadata-document/

# Client ID Metadata Document




```
{
  "client_id": "https://example.com/client.json",
  "client_name": "Example Client",
  "client_uri": "https://example.com",
  "logo_uri": "https://example.com/logo.png",
  "redirect_uris": [
    "https://example.com/redirect"
  ]
}
```

The URL is used in an Authorization request

`/authorize?client_id=https://example.com/client.json&...`

https://datatracker.ietf.org/doc/draft-parecki-oauth-client-id-metadata-document/

# Applicability

Pre-registration of clients is not possible when the client developer has no prior relationship with the authorization server, for example:

- Open source chat app connecting to self-hosted chat server
- Apps that connect to self-hosted services, such as Mastodon or WordPress, where there is no single central server to register the OAuth client
- “Federations” where entities can join through a trust chain but are not all pre-registered with each other.
- Agentic AI!

# The Potential Problem

There may be multiple ways for an AS to interpret a URL as a client identifier:

OpenID4VP `x509_san_uri`, OpenID4VP `redirect_uri`, OpenID4VP over the Digital Credentials API, OpenID Federation, W3C IndieAuth, and Client Metadata Document all use `https` URLs as the Client ID value

The resulting problems are:

- A generic AS doesn't know how to resolve metadata, and the client has no way to indicate how it expects the AS to resolve
- Potential attacks are possible if a malicious client has the ability to control the document pointed to by the URL of another client when interpreted through a different mechanism

# Why not add a new parameter?

`/authorize?client_id=https://example.com/metadata.json&client_id_scheme=foo…`

This was previously proposed in OpenID4VP, but eventually determined it would cause more problems than it would solve.

https://github.com/openid/OpenID4VP/issues/124

(For example, how would this be extended for RFC9068 access tokens? Defining the `client_id_scheme` there may also lead to confusion attacks for systems that don’t support the parameter. See the thread above for more.)

# Why not define a new parameter for each type?

e.g. `client_url` instead of `client_id`

`/authorize?`**`client_url`**`=https://example.com/metadata.json...`

- Breaks compatibility with OAuth, since `client_id` would not be present anywhere
- All client and server libraries would need to be updated to support the new parameter and not require the old parameter
- Would need to define a new claim in RFC 9068 access tokens
- Anywhere client ID is used elsewhere would need to be renamed to avoid the same confusion attacks (e.g. the `aud` claim in OpenID Connect ID tokens)
- Each client ID method would need its own parameter instead of `client_id`: `client_url=, did=, verifier_attestation=, ...`

# Client ID Prefix in OpenID for Verifiable Presentations

https://openid.net/specs/openid-4-verifiable-presentations-1_0.html#section-5.9

As of 4VP PR #401 requires a prefix for every value, no fallback exception for https or did.

```
<client_id_prefix>:<client_id_value>
```

```
"redirect_uri:https://client.example.org/cb",
"decentralized_identifier:did:example:123#1",
"verifier_attestation:verifier.example",
"x509_san_dns:client.example.org",
"x509_hash:Uvo3HtuIxuhC9[...]viRiA0OZszk",
"origin:https://verifier.example.com"
```

prefix value

https://github.com/openid/OpenID4VP/pull/401#issuecomment-2648740662

# Default Client Identifier Prefix

“If a `:` character is present in the Client Identifier but the value preceding it is not a recognized and supported Client Identifier Prefix value, the Authorization Server MAY treat the Client Identifier as having a default Client Identifier Prefix.”

Enables plain URLs as Client IDs for existing OpenID Federation deployments and Client ID Metadata Documents.

# Proposal

Call for adoption?
https://datatracker.ietf.org/doc/draft-parecki-oauth-client-id-prefix/

With the allowance for default Client ID Prefix as previously described, an AS can use only one default Client ID Prefix.

This creates the building block that can then be used by

- OpenID Federation
  - https://openid.net/specs/openid-federation-1_0.html
- OpenID for Verifiable Presentations
  - https://openid.github.io/OpenID4VP/openid-4-verifiable-presentations-wg-draft.html
- Client Metadata Documents
  - https://datatracker.ietf.org/doc/draft-parecki-oauth-client-id-metadata-document/

Then, update Client ID Metadata Documents to reference the Client ID Prefix draft.