What if a `client_id` is not managed by a (trusted) 3rd party?

Kristina Yasuda

`client_id` as defined in RFC6749

2.2. Client Identifier

The authorization server issues the registered client a client identifier -- a unique string representing the registration information provided by the client. The client identifier is not a secret; it is exposed to the resource owner and MUST NOT be used alone for client authentication. The client identifier is unique to the authorization server.

The client identifier string size is left undefined by this specification. The client should avoid making assumptions about the identifier size. The authorization server SHOULD document the size of any identifier it issues.

Requirements

- AS has to issue client identifier to the Client.

Also Section 2:

"Before initiating the protocol, the client registers with theauthorization server."

Limitations faced

1. Example OpenID 4 Verifiable Credential Issuance
    - Large number of Wallets and large number of Credential Issuers that may interact with each other - unrealistic/impractical for the Wallets to separately register with every issuer and manage `client_id` for each Issuer.
    - High assurance (e.g. eIDAS v2): 3rd parties manage wallet trust list (similar to open banking)
    - Anonymous Clients / Open model: some Issuers want to enable issuance into the wallets without requiring those Wallets to have a client id at the Issuer
2. Other examples of closed ecosystem, where only group of Clients that meet specific requirements can interact with a group of AS
    - Open Banking
    - EDU federation

The Questions

- Who determines the client id?
- Who manages the client metadata?

(Some data points…)

`client_id`-as-a-URI model as defined in OpenID Federation

The specification is interesting for two reasons:

1. A third party manages the client data and the client does NOT register (in automatic mode)
2. The client determines its client id.

It describes how two entities that would like to interact can dynamically fetch and resolve trust and metadata for a given protocol through the use of third-party Trust Anchor.


```
{
  "iss": "https://feide.no",
  "sub": "https://ntnu.no",
  "iat": 1516239022,
  "exp": 1516298022,
  "crit": ["jti"],
  "jti": "7l2lncFdY6SlhNia",
  "policy_language_crit": ["regexp"],
  "metadata": {
    "openid_provider": {
      "issuer": "https://ntnu.no",
      "organization_name": "NTNU",
    },
    "oauth_client": {
      "organization_name": "NTNU"
    }
  },
  "metadata_policy": {
    "openid_provider": {
      "id_token_signing_alg_values_supported":
        {"subset_of": ["RS256", "RS384", "RS512"]},
      "op_policy_uri": {
        "regexp":
          "^https:\/\/[\\w-]+\\.example\\.com\/[\\w-]+\\.html"}
    },
    "oauth_client": {
      "grant_types": {
        "subset_of": ["authorization_code", "client_credentials"]},
      "scope": {
        "subset_of": ["openid", "profile", "email", "phone"]}
    }
  },
  "constraints": {
    "max_path_length": 2
  },
  "jwks": {
    "keys": [
      {
        "alg": "RS256",
        "e": "AQAB",
        "key_ops": ["verify"],
        "kid": "key1",
        "kty": "RSA",
        "n": "pnXBOusEANuug6ewezb9J_...",
        "use": "sig"
      }
    ]
  }
}
```

`client_id`-as-a-URI model as defined in SIOPv2 and OpenID4VP

New options to allow Just-in-time metadata exchange to enable deployments models where the AS does not or cannot support pre-registration of Client metadata.

- `client_id` equals `redirect_uri`
- OpenID Federation 1.0 Automatic Registration.
- Decentralized Identifiers with `client_metadata` or `client_metadata_uri` containing RP Registration Metadata in the Authorization Request

※ In addition to an option for the AS to obtains Client metadata prior to a transaction, e.g using RFC7591 or out-of-band mechanisms.

Anonymous clients

- The OpenID Connect WG discusses to allow for anonymous access to the AS in case of the pre authorized grant type in OpenID4VCI, when no Authotization Endpoint involved.
  - JWT bearer grant type already supports anonymous clients

(other very interesting work arounds also emerging…)

Why is `client_id` needed?

1. Tying client_id with a list of allowed redirect_uris, to prevent open redirects.
2. Knowing Client’s registration metadata
3. Knowing metadata to display to the Users
4. AS applying specific permissions based on client_id

Next Steps?

- Drop the requirement for RFC6749 section 2.2 for the AS to issue a `client_id`
- Can we enable `client_id`s not managed by a (trusted) 3rd party?
  - Some mechanisms already defined/proposed: OpenID Federation specification (Entity Statements), draft-looker-oauth-client-discovery-01, etc.
  - Loosen the requirement that the AS has to issue client identifier to the Client.