Managing authorization grants beyond OAuth 2.0

OID 2021

Fabien Imbault, Justin Richer and Aaron Parecki
What this paper is about

- Review pros and cons of OAuth2

Why we’re working on GNAP

- IETF GNAP Grant Negotiation and Authorization Protocol
  - You’re welcome to participate
    - Join the mailing list https://datatracker.ietf.org/wg/gnap/documents/
    - Participate in issues https://github.com/ietf-wg-gnap/gnap-core-protocol
A primer


Resource Server (RS) = where there are protected resources, that require authorization to allow access (under the form of an access token)

Resource Owner (RO) = who owns the resource

End-user = who requires access through a client

In many cases: RO = end-user (ex: access to my banking account through a mobile app)

But not always: RO (patient) != end-user (doctor)
Beyond the web browser

- Web browser is only one interaction method amongst other

```
"interact": {

  "start": ["redirect", "user_code", "app"],

  "finish": {

    "method": "redirect",

    "uri": "https://client.example.net/return/123455",

    "nonce": "LKLTI25DK82FX4T4QFZC"

  }

}
```
Negotiation

- Interact / Continue API
Client instance

- Instead of registered client ID

```
"client": {
  "key": {
    "proof": "httpsig",
    "jwk": { ... },
    "cert": "MIIEHDCCAwSgAwIBAgIBATANBgkqhkiG9w0BAQsFA...
  },
  "class_id": "web-server-1234",
  "display": { "name": "My Client Display Name", "uri": "https://example.net/client" }
}
```
Subject identifier

- Support for various identifier formats (opaque, DID, etc.) and assertions (idtoken, saml2)


```
"subject": {
    "sub_ids": [ {
        "format": "opaque",
        "id": "J2G8G8O4AZ"
    } ],
    "assertions": {
        "id_token": "eyj..."
    }
}
```

- GNAP aims direct support of OIDC but also SSI (cf “AS as a token” model)
Expanded delegation

- Richer request (aligned with RAR), support ACLs and capabilities

"access": [

{

  "type": "photo-api",

  "actions": ["read", "write", "delete"],

  "locations": ["https://server.example.net/", "https://resource.local/other"],

  "datatypes": ["metadata", "images"],

  "privileges": ["admin"],

}]

]
Security

- Prove possession of key / rotate keys
- Various mechanisms, such as JWS, mTLS, httpsig


- Contributions on threats and security considerations welcome!
Privacy

- GNAP tries to limit the odds of a consolidation to just a handful of super-popular AS services

- Additional spec to deal with AS-RS relationships
  - [https://github.com/ietf-wg-gnap/gnap-resource-servers](https://github.com/ietf-wg-gnap/gnap-resource-servers)
  - Ex: delegation tokens
AS as Token Factory

Authenticate
Authorize
Delegate

Method details
(Out of scope)

“Interaction Method”

End User

Request Token

Issue Token

Use Token

Client Instance

Validate/Trust Token

AS

RS
AS as Token Factory

Authenticate
Authorize
Delegate

Request Token

Issue Token

Use Token

Authenticate
Authorize
Delegate

Authenticate
Authorize
Delegate

Issue Token

Use Token

Client Instance

End User

AS

RS

RO
AS and RS Relationship

Authenticate
Authorize
Delegate

Request Token
Issue Token
Use Token

Client Instance

GNAP Core

AS

Validate/Trust Token
Register Resources
Decide Formats

RS

GNAP RS
Additional resources


- There is a longer version of the paper at https://blog.fimbault.com/managing-authorization-grants-beyond-oauth-2

- Things you can’t do well in OAuth2
  - We cover some examples that would be impossible to do in OAuth2 / UMA2 (medical team)
  - Through a NGI_TRUST grant, we also extended GNAP to cover IoT scenarios https://blog.fimbault.com/lessons-learned-from-our-medium-project