Grant Negotiation and Authorization Protocol (GNAP)

Justin Richer

Open Payments WG 2023
This is not an extension of OAuth 2
This is not “OAuth 3”
If we were building the OAuth ecosystem today, what would it look like?
Design Pillars

- Protocol for negotiating access
- Methods for interacting with humans
- Validating and verifying the client software
- Methods for binding keys to message requests
- Data model of what’s being requested
Explicit Grant Request Lifecycle

- Client instance creates a grant request
- Resource owner authorizes a grant request
- Authorization server issues an access token from a grant request
Negotiating Access
GNAP Allows Conversation

- Client instance asks for what it wants and presents what it knows
- AS responds based on that request
- Conversation can continue over time and be augmented by both parties
- Users can get involved when needed
GNAP Request

```json
{
  "access_token": {
    "access": [ "dolphin-metadata", "and another thing" ]
  },
  "client": "xyz-client-1234a",
  "interact": {
    "start": [ "redirect", "app" ],
    "finish": {
      "method": "redirect",
      "uri": "https://client.example.net/return/123455",
      "nonce": "LKLTI25DK82FX4T4QFZC"
    }
  }
}
```
GNAP Request (Complex API)

```json
{
    "access_token": {
        "access": [
            "foo",
            "bar",
            {
                "type": "example.com/resource-set",
                "actions": ["read", "write", "dolphin"],
                "locations": ["https://server.example.net/",
                              "https://resource.local/other" ],
                "datatypes": ["metadata", "images" ]
            },
            "dolphin-metadata"
        ],
        "dolphin-metadata"
    },
    "client": "xyz-client-1234a",
    "interact": ...
}
```
GNAP Request (Dynamic Client)

```json
{
    "access_token": {
        "access": [ "dolphin-metadata", "and another thing" ]
    },
    "client": {
        "key": {
            "proof": "httpsig",
            "jwk": { "kty": "RSA", "e": "AQAB", ... }
        },
        "display": {
            "name": "My Client Display Name",
            "uri": "https://example.net/client"
        }
    },
    "interact": ...
}
```
GNAP Response

```json
{
    "interact": {
        "start": {
            "redirect": "https://server/interact/4CF492KHQ"
        },
        "finish": "MBDOFXG4Y5CVJJCX821LH"
    },
    "continue": {
        "access_token": "80UPRY5NM33OMUKMKSU",
        "uri": "https://server.example.com/continue",
        "wait": 60
    }
}
```
GNAP Response (Access Token)

```
{
    "access_token": {
        "value": "I25DK82FX4TI25DK8225DK82FX4T4QFZC",
        "access": [ "dolphin-metadata", "and another thing" ]
    },
    "subject": {
        "sub_ids": [ { "format": "opaque", "id": "J2G8G804AZ" } ],
        "updated_at": "2020-01-01T12:43:29+0000"
    }
}
```
What This Means

- No need for complex pre-configuration
  - Discovery, registration, extension all happen in-line
  - Sensible and predictable failure states
- All requests start the same way
- Systems can adapt at runtime based on what’s possible and what’s needed
1. Start Session
2. Request Access
3. Interaction Needed
4. Redirect for Interaction
5. AuthN
6. AuthZ
7. Redirect for Continuation
8. Continue Request
9. Grant Access
10. Access API
11. API Response
Interacting With Users
Negotiating GNAP Interaction

Ways to Interact With the User

Start Interaction
- Redirect to URL
- User Code
- Application Launch
- Some Extension

Finish Interaction
- Method: Redirect
Negotiating GNAP Interaction

Ways to Interact With the User

Start Interaction
- Redirect to URL
- User Code
- Application Launch
- Some Extension

Finish Interaction
- Method: Redirect

Ways to Interact With the User

Start Interaction
- Redirect to URL
- User Code
- Application Launch
- Some Extension

Finish Interaction
- Method: Redirect
GNAP Interaction

• Client Instance declares **what it can do**

• AS chooses **from that set** based on:
  — What AS can support
  — What is needed for the request

• No interaction needed? Just return results

• Can't support what the client can do? No interaction
AS is a Token Factory
Factories take **raw material** and produce consumable products
Token Factory Pattern

- Allows innovation in claims presentation at the AS
  - AS can face the distributed/federated world
  - Doesn't assume user with account
- Encourages RS to closely trust its AS
- Doesn’t assume deployment patterns for AS
- Aligns with deployed reality of OAuth 2 and UMA2
Client Software
Client Instances in GNAP

- Client instances are identified by keys
  - Some keys are ephemeral
  - Some keys are pre-registered
  - Instance identifiers can act as a shorthand reference
- Protocol does not need client identifiers
  - Differentiates between client instance and client software
Key proofs in GNAP

• All messages to the AS are signed by the client
  – Starting request, continuation requests

• Access tokens are key-bound by default
  – I.e.: messages to the RS are also signed by the client, using the same methods as when talking to the AS

• Flexible signature methods
  – HTTP Message Signatures, MTLS, JOSE, …
Key Binding
Unsigned GNAP Request

POST /gnap HTTP/1.1
Host: server.example.com
Content-Type: application/json
Content-Length: 986
Content-Digest: sha-256=:98QzyNVYpdgTrWBKpC4qFSCmmR+CrwwvUoiaDCSjKxw=: 

```json
{
   "access_token": ...
   "client": ...
}
```
"@method": POST
"@target-uri": https://server.example.com/gnap
"content-type": application/json
"content-digest": sha-256=:98QzyNVYpdgTrWBKpC4qFSCmmR+CrwwvUoiaDCSjKxw=: 
"content-length": 986
"@signature-params": ("@method" "@target-uri" "content-type" "content-digest" "content-length");created=1618884475;keyid="gnap-rsa"
POST /gnap HTTP/1.1
Host: server.example.com
Content-Type: application/json
Content-Length: 986
Content-Digest: sha-256=:98QzyNVYpdgTrWBKpC4qFSCmmR+CrvwwvUoiaDCSjKxw=:
Signature-Input: sig1=("@method" "@target-uri" "content-type"
"content-digest" "content-length");created=1618884475
;keyid="gnap-rsa"
Signature: sig1=:H4110tgBS6vdGGxKkRGfKSmpZG0vU6lp1V9cDeZyy9o8fscqSLIynpTVLZ
Gv0/bStSoiLN6openFNRAldgopGfhLH8zk09SF/H3frb9b0PkV+/4iuh/ENJ9jeIg+UupLknP
qvghBmhFcdNw1cQRtR4SF1E2KNaI93owwL2+2r9q1lzBoWiKqWfiTCshB49k50kzf78JGYJku
8iubNwhT55ULJNYy0s7HVm50EJe2VtCfnPQ27yoCpzBwUx7DMlP5W5ioaK7iaJFFqZQ5Jjrq
4f8ZH MnayEaMQ9oKDX0/HLY0qvgrzuT7T1zJV5q0qe4J7909sXL30YQ5cmndUV8FC==:=
{
"access_token": ...
"client": ...
}
HTTP Message Signature (with Token)

POST /foo?param=value&pet=dog HTTP/1.1
Host: example.com
Date: Tue, 20 Apr 2021 02:07:55 GMT
Content-Type: application/json
Content-Length: 18
Authorization: HTTPSig 3ZM-BOXPQTR31UOH6XKG.WEM1N3G98L

Signature-Input: sig1="@method" "@authority"
"content-type" "authorization";created=1618884475;keyid="test-key-rsa-pss"

Signature:
sig1=:
1Zg7ZGYD+ngJyVn805r73rh2eFCPO+ZXDs45Is/Ex8srzGC9sfVZfqeEfApRFFe5yXDMANVUwzFWCEnGM6+SJVvmW1l/jyEn45qaA6Hw+ZDHbrbp6qvD4N0S92j1PyVVEh/SmCwnkeNiBgnbt+E0K5wCFNHPbo4X1Tj406W+bTtnKzaokxXbWkW8aIQ7rg92zqE1oqBRjqtRi5/Q6P5ZYYGGINKzNyV3UjZtxeZNNNJ+MANwS0mofFqcZHVGsSU/1wUzP7MhzOKLca1Yg==:

{"hello": "world"}
What’s Being Requested
Types of Data

- APIs the Client Wants to Access
- Information the Client Wants About the User
- Information the Client Has About the User
Types of Data

- APIs the Client Wants to Access
- Information the Client Wants About the User
- Information the Client Has About the User

AS -> C
AS -> C
C -> AS
API Access
GNAP Access Token Request

```
"access_token": {
    "access": [
        {
            "type": "photo-api",
            "actions": ["read", "write", "dolphin"],
            "locations": [
                "https://server.example.net/",
                "https://resource.local/other"
            ],
            "datatypes": ["metadata", "images"]
        },
        "read"
    ]
}
```
Two equivalent requests

Object:
"access": [
  {
    "type": "photo-api",
    "actions": [ "read" ],
    "datatypes": [ "metadata" ]
  },
  {
    "type": "photo-api",
    "actions": [ "write" ],
    "datatypes": [ "image-data" ]
  }
]

String:
"access": [
  "metadata",
  "update-image"
]

The AS decides how this is mapped
Requesting Multiple Access Tokens

"access_token": [
    {
        "access": [ "write", "blow-up" ],
        "label": "danger-token"
    },
    {
        "access": [ "read", "think" ],
        "flags": [ "bearer" ],
        "label": "other-token"
    }
]
Receiving Multiple Access Tokens

```
"access_token": [
    {
        "access": ["write", "blow-up"],
        "label": "danger-token",
        "value": "7Z5QBH2UR9.5T3F5I6NM7"
    },
    {
        "access": ["read", "think"],
        "flags": ["bearer"],
        "label": "other-token",
        "value": "E5D1R95TMX-KCG2ZIXHK0"
    }
]
```
Subject Information

- Represents the “current user”
- Available as assertions or identifiers
  - GNAP does not define formats for either
- Requested separately from API access
  - Passes directly between the AS and client
  - Not tied to an access token
  - No separate API to call
GNAP Subject Information

```json
{
  "subject": {
    "sub_ids": [
      {
        "format": "opaque",
        "id": "J2G8G8O4AZ"
      }
    ],
    "updated_at": "2020-01-01T12:43:29+0000"
  }
}
```
Sending user information

• Sometimes the client knows who the user is
• GNAP allows the client to send that information to the AS
• The AS decides how much to trust that
  – Sometimes, interaction can be skipped
Principle: It's not just APIs

- OpenID Connect showed us how to add identity to a delegation API
  - You're delegating access to the identity
- GNAP builds this core concept in
  - Does not define a full identity API with schemas, endpoints, etc.
Thank you!

taxauth@ietf.org