Grant Negotiation and Authorization Protocol (GNAP)

Justin Richer

Open Payments WG 2023

@justin _ _ richer

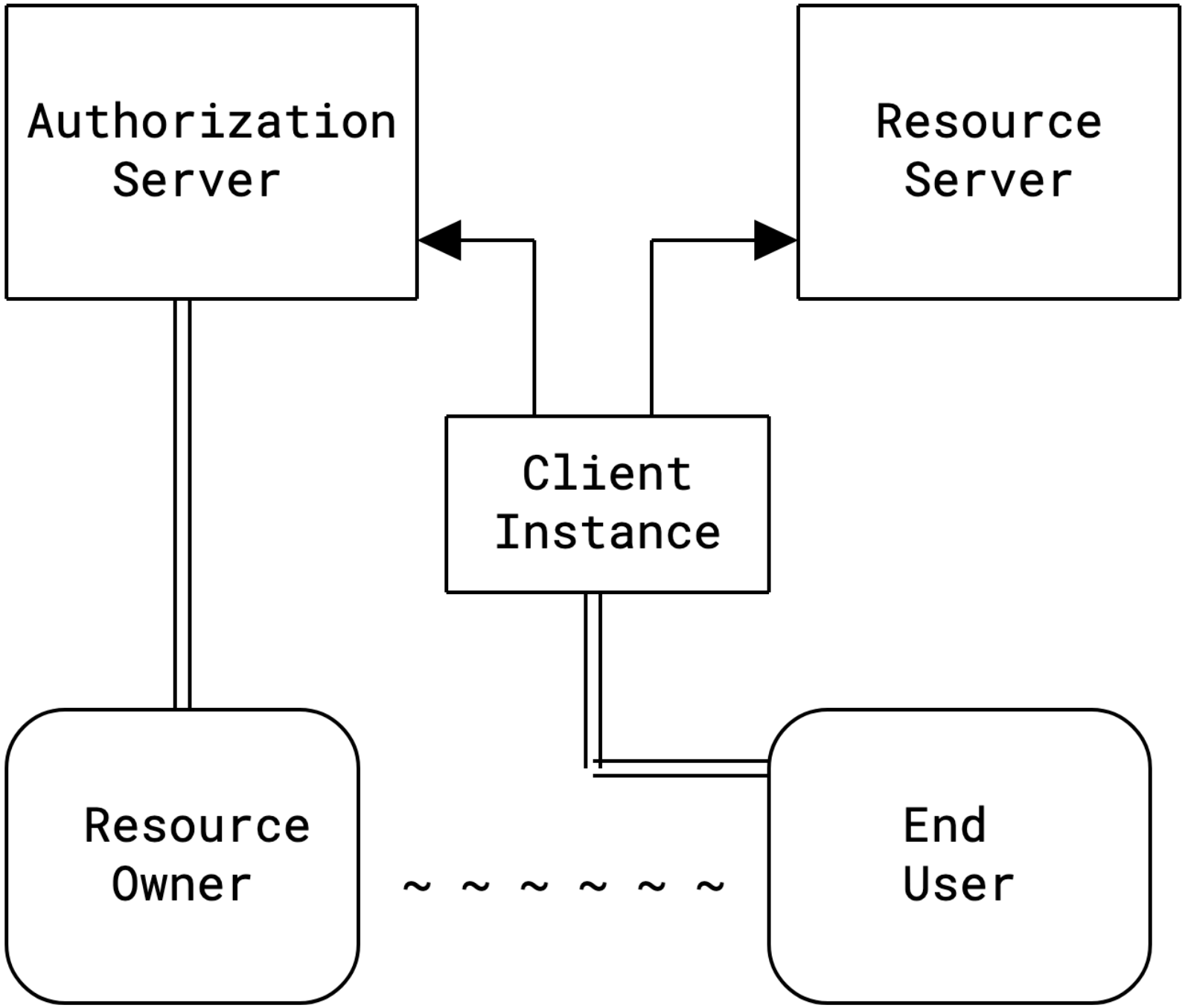
This is not an extension of OAuth 2

This is not "OAuth 3"

If we were building the OAuth ecosystem today, what would it look like?

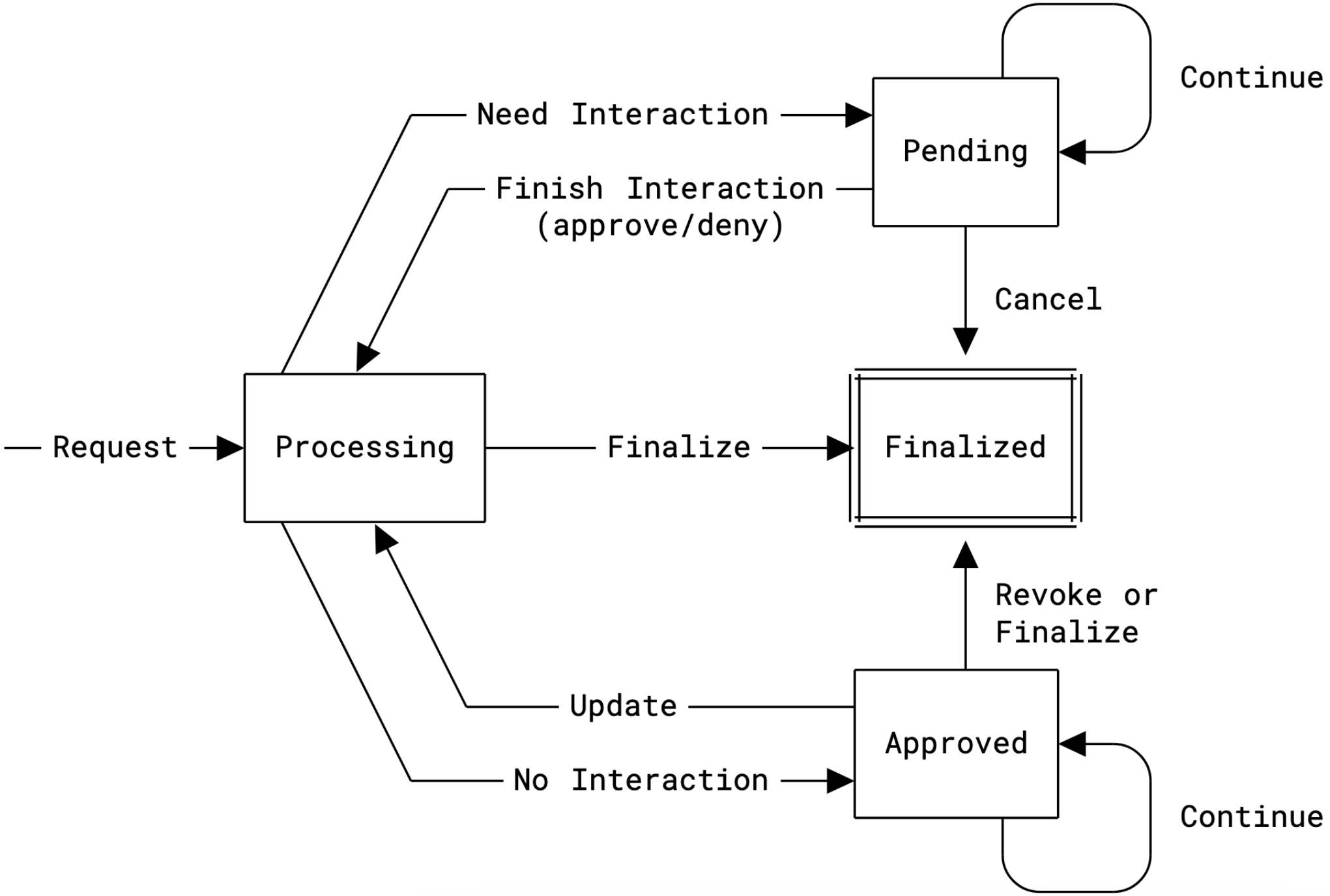
Design Pillars

- 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


Authorization Server
Resource Server
Client Instance
Resource Owner
~ ~ ~ ~ ~ ~
End User

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


Continue
Need Interaction
Pending
Finish Interaction
(approve/deny)
Cancel
Request
Processing
Finalize
Finalized
Revoke or
Finalize
Update
Approved
No Interaction
Continue

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

```
{
  "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"
    }
  }
}
```

API Access

Identifying the client

Starting interaction

Finishing interaction

GNAP Request (Complex API)

```
{
  "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"
    ]
  },
  "client": "xyz-client-1234a",
  "interact": ...
}
```

API Access

GNAP Request (Dynamic Client)

```
{
  "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": ...
}
```

Client instance key

Client instance display information

GNAP Response

```
{
  "interact": {
    "start": { "redirect": "https://server/interact/4CF492KHQ" },
    "finish": "MBDOFXG4Y5CVJCX821LH”
  },
  "continue": {
    "access_token": { "value": "80UPRY5NM33OMUKMKSKU" },
    "uri": "https://server.example.com/continue",
    "wait": 60
  }
}
```

Handle Interaction

Continue the request

GNAP Response (Access Token)

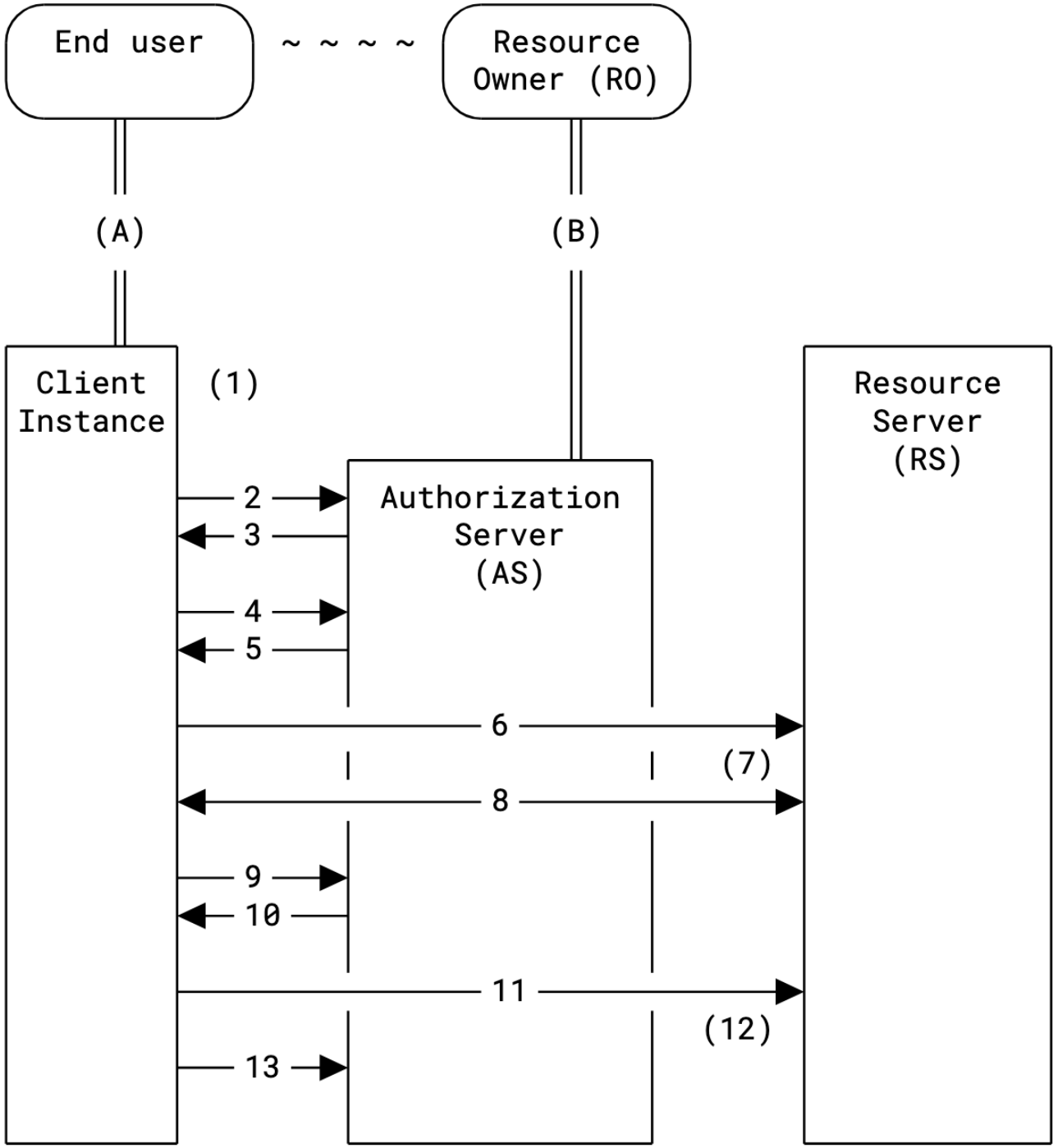
```
{
  "access_token": {
    "value": "I25DK82FX4TI25DK8225DK82FX4T4QFZC",
    "access": [ "dolphin-metadata", "and another thing" ]
  },
  "subject": {
    "sub_ids": [ {
      "format": "opaque",
      "id": "J2G8G8O4AZ"
    } ],
    "updated_at": "2020-01-01T12:43:29+0000"
  }
}
```

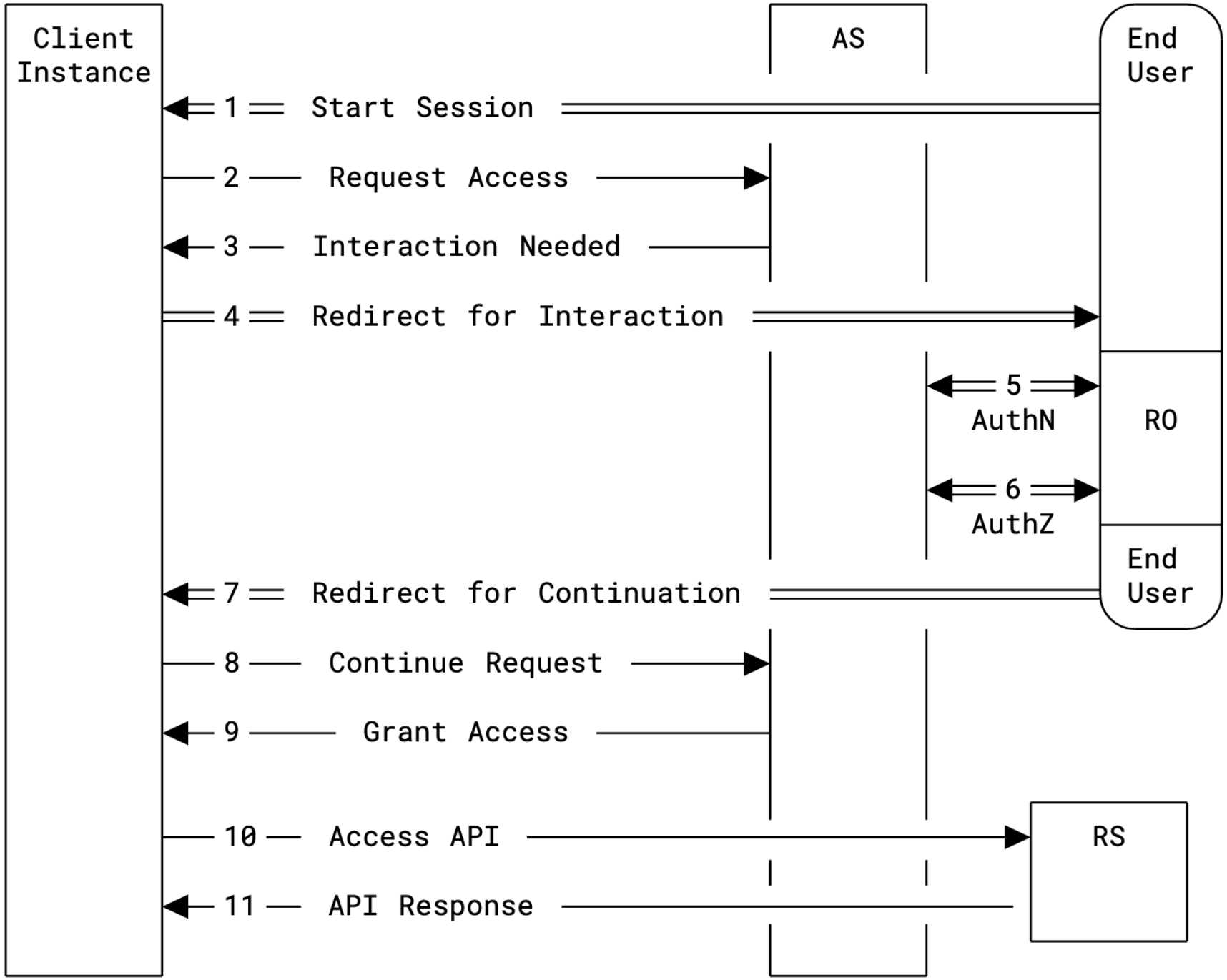
API Access

Subject identification

What This Means

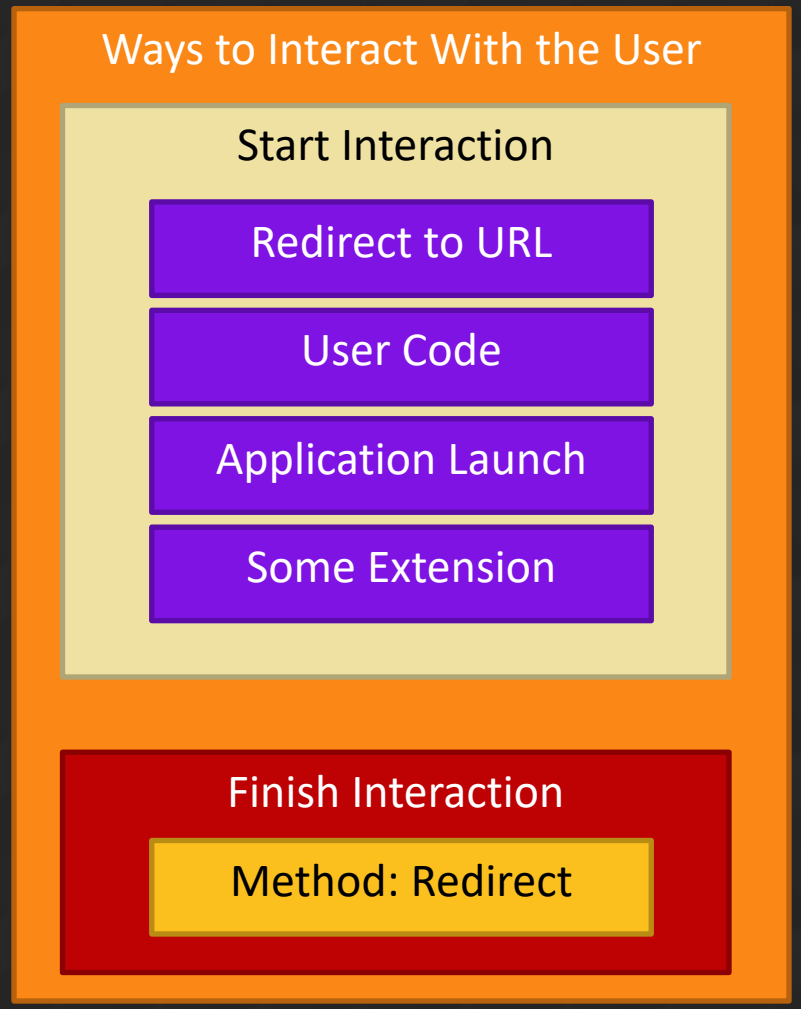
- No need for complex pre-configuration
  - Discovery, registration, extension all happen in-line
  - Sensible and predictable failure states
- All requests start the same way
- Systems can adapt at runtime based on what's possible and what's needed


End user
~ ~ ~ ~
Resource Owner (RO)
(A)
(B)
Client Instance
(1)
Authorization Server (AS)
Resource Server (RS)
2
3
4
5
6
(7)
8
9
10
11
(12)
13

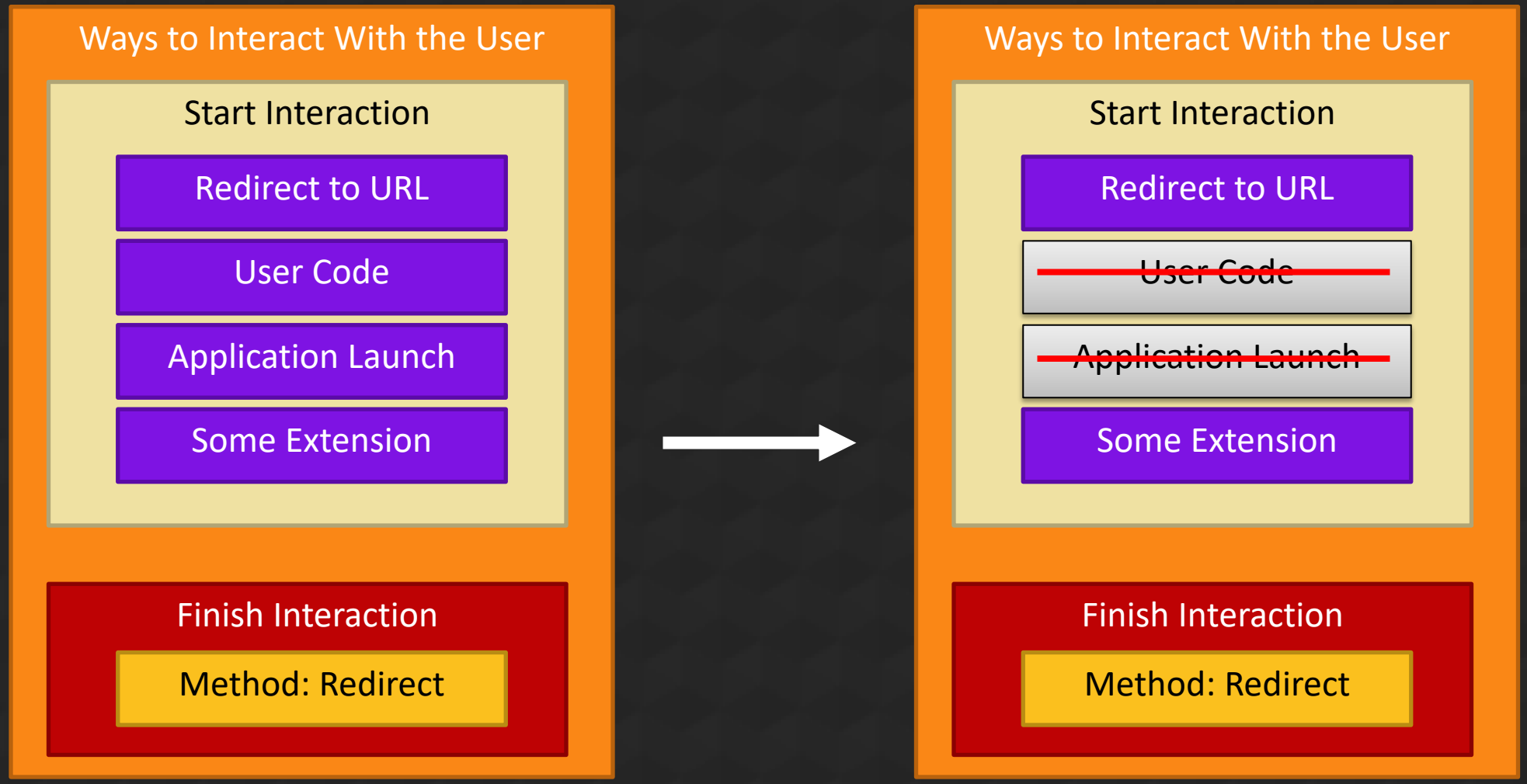

Client Instance
AS
End User
1 Start Session
2 Request Access
3 Interaction Needed
4 Redirect for Interaction
5 AuthN
RO
6 AuthZ
End User
7 Redirect for Continuation
8 Continue Request
9 Grant Access
10 Access API
RS
11 API Response

Interacting With Users

Negotiating GNAP Interaction



Negotiating GNAP Interaction

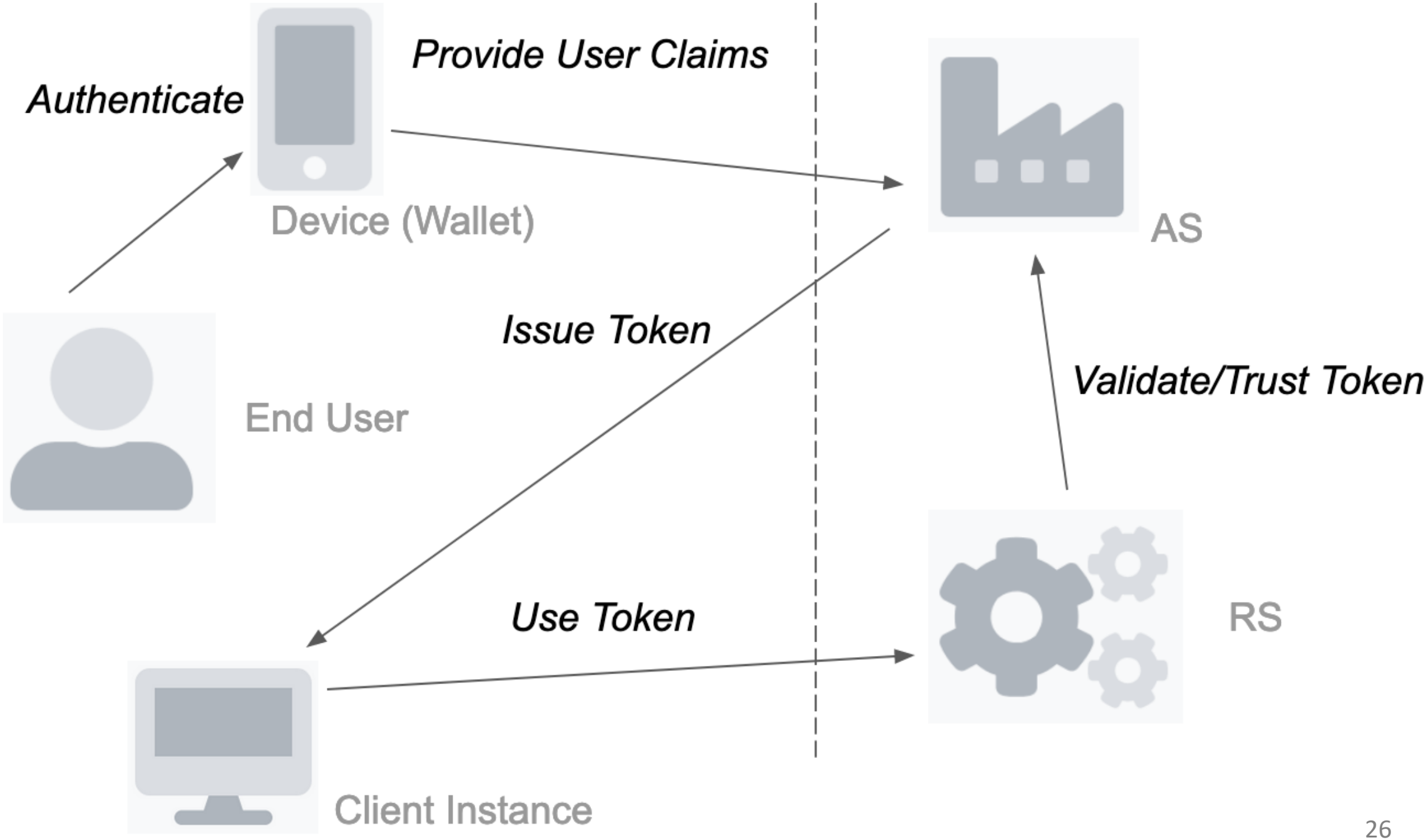


GNAP Interaction

- 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

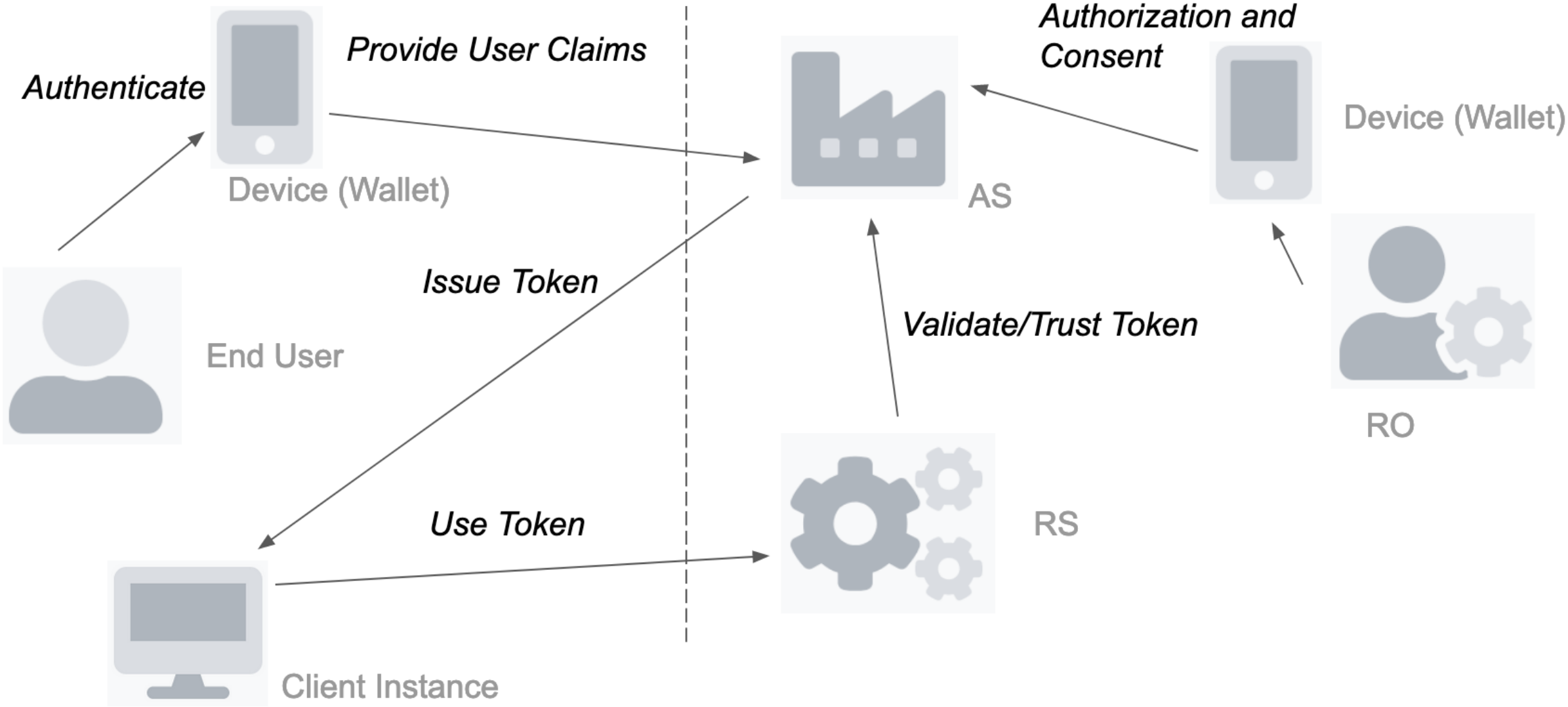
AS is a Token Factory

Factories take raw material
and produce consumable products


Authenticate
Provide User Claims
Device (Wallet)
AS
Issue Token
Validate/Trust Token
End User
Use Token
RS
Client Instance

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


Authenticate
Provide User Claims
Device (Wallet)
Authorization and Consent
Device (Wallet)
AS
End User
Issue Token
Validate/Trust Token
RO
Use Token
RS
Client Instance

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=:

{
 "access_token": ...
 "client": ...
}
```

Signature Base

```
"@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"
```

Signed GNAP Request

```
POST /gnap HTTP/1.1
Host: server.example.com
Content-Type: application/json
Content-Length: 986
Content-Digest: sha-256=:98QzyNVYpdgTrWBKpC4qFSCmmR+CrwwvUoiaDCSjKxw=:
Signature-Input: sig1=("@method" "@target-uri" "content-type"
 "content-digest" "content-length");created=1618884475
 ;keyid="gnap-rsa"
Signature: sig1=:H411OtgBS6vdGGxKkRGfKSmZG0vU6lp1V9cDeZyy9o8fscqSLIynpTVLZ
 GvO/bStSoiLNa6penFNRAldgopGfhlH8zk09SF/H3frb9b0PkV+/4iuh/ENJ9jeIg+UupLn9P
 qvghBmhFcdNwlcQRtR4SF1E2KNaI93owwl2+2r9q1lzBoWiKqWfiTCshB49k5Okzf78JGYJku
 8iubNwhT55UlJNYy0s7HVm50EJe2VtCfnPQ27yoCpzBwUx7DMlP5W55ioaK7iaJFFqZQ5Jjrq
 4f8ZHMnayEaMQ9oKDXO/HLY0qvgrzuT7T1zJV5q0qe4J79O9sXL30YQ5cmnduV8FCA==:

{
 "access_token": ...
 "client": ...
}
```

@justin_richer

https://bspk.io/

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-B0XGPQTR31UOH6XKG.WEM1N3G98L
Signature-Input: sig1=("@method" "@authority"
  "content-type" "authorization");created=1618884475;keyid="test-key-rsa-pss"
Signature:
sig1=:NtIKWuXjr4SBEXj97gbick4O95ff378I0CZOa2VnIeEXZ1itzAdqTpSvG91XYrq5CfxCmk8zz
1Zg7ZGYD+ngJyVn805r73rh2eFCPO+ZXDs45Is/Ex8srzGC9sfVZfqeEfApRFFe5yXDmANVUwzFWCEn
GM6+SJVmWl1/jyEn45qA6Hw+ZDHbrbp6qvD4N0S92jlPyVVEh/SmCwnkeNiBgnbt+E0K5wCFNHPbo4X
1Tj406W+bTtnKzaoKxBWKW8aIQ7rg92zqE1oqBRjqtRi5/Q6P5ZYYGGINKzNyV3UjZtxeZNnNJ+MAnW
S0mofFqcZHVgSU/1wUzP7MhzOKLca1Yg==:

{"hello": "world"}
```

What's Being Requested

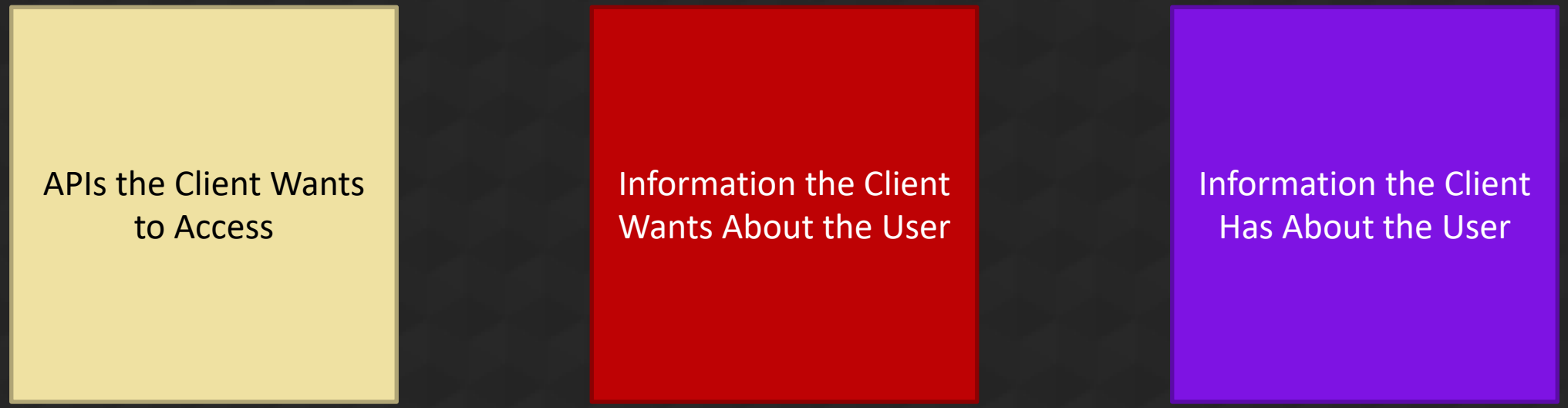
Types of Data

APIs the Client Wants to Access

Information the Client Wants About the User

Information the Client Has About the User

Types of Data



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"
    ]
}
```

Type of API

What you're doing

Where you're doing it

What you're doing it to

Reference shortcut

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"
  }
]
```

Bound token with destructive access

Bearer token with weaker access

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-KCG2ZIXHKO"
  }
]
```

Bound token with destructive access

Bearer token with weaker access

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

```
{
  "subject": {
    "sub_ids": [
      {
        "format": "opaque",
        "id": "J2G8G804AZ"
      }
    ],
    "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!

txauth@ietf.org