# HTTP API Authorisation

## Building on OAuth2.0 and OpenID Connect
## Experiences at UiO's Services for Sensitive Data

**Leon du Toit - Section Leader, Research Platforms, UiO-IT**

# Overview

1. The problem
2. The solution
3. Experience at UiO/TSD
4. Recommendations

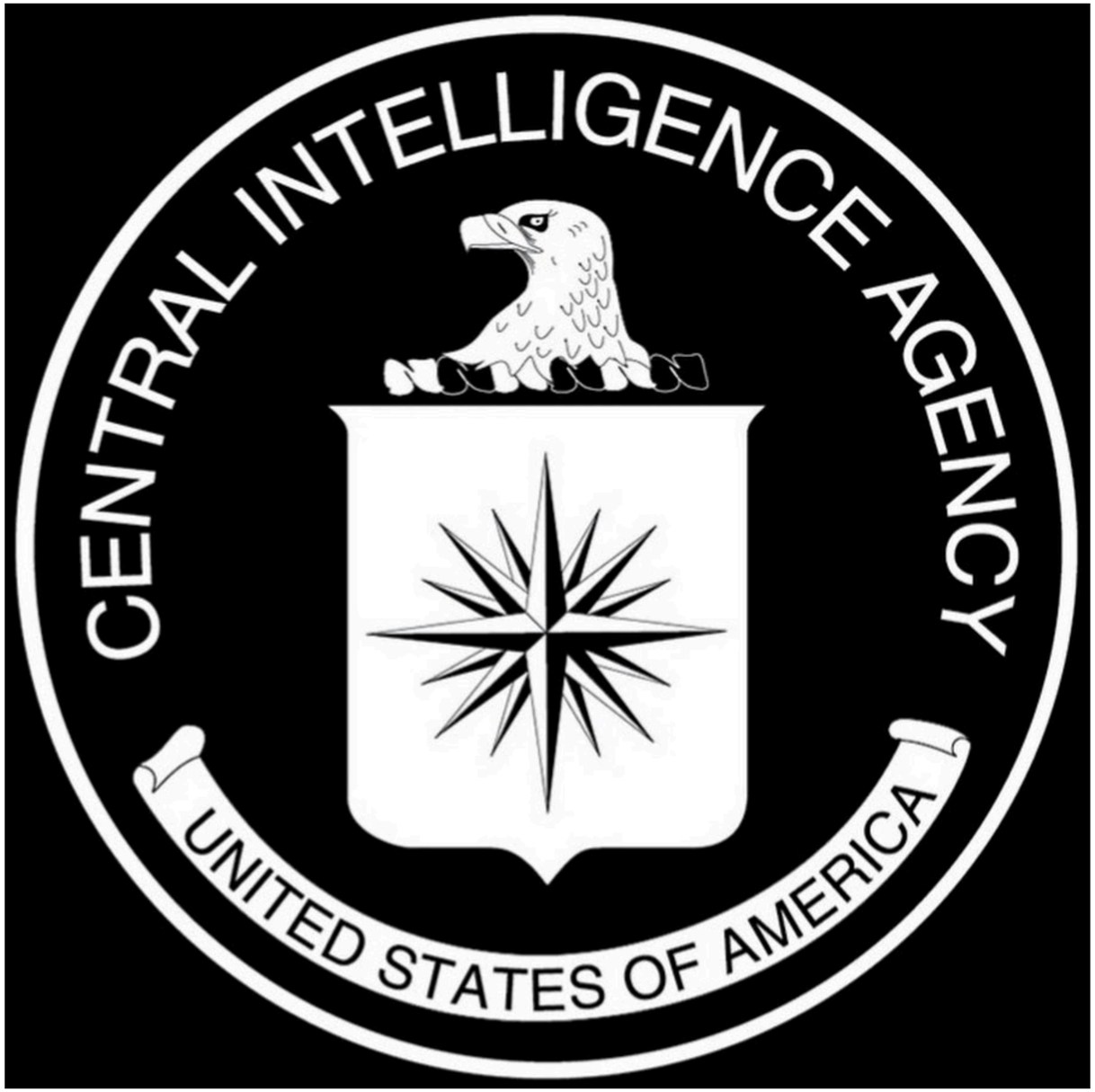
# HTTP Request

**rfc2616#5, rfc9113#4, rfc9114#5**

GET /resource?query=…

Host: api.service.no

Authorization: Bearer $JWT


CENTRAL INTELLIGENCE AGENCY
UNITED STATES OF AMERICA

# CIA

## HTTP API request authorisation

- **Confidentiality**
- Integrity
- **Availability**

# HTTP Request

**rfc2616#5, rfc9113#4, rfc9114#5**

GET /resource?query=…

Operation, Resource

Host: api.service.no

Authorization: Bearer $JWT

# HTTP Request

**rfc2616#5, rfc9113#4, rfc9114#5**

GET /resource?query=…

Host: api.service.no

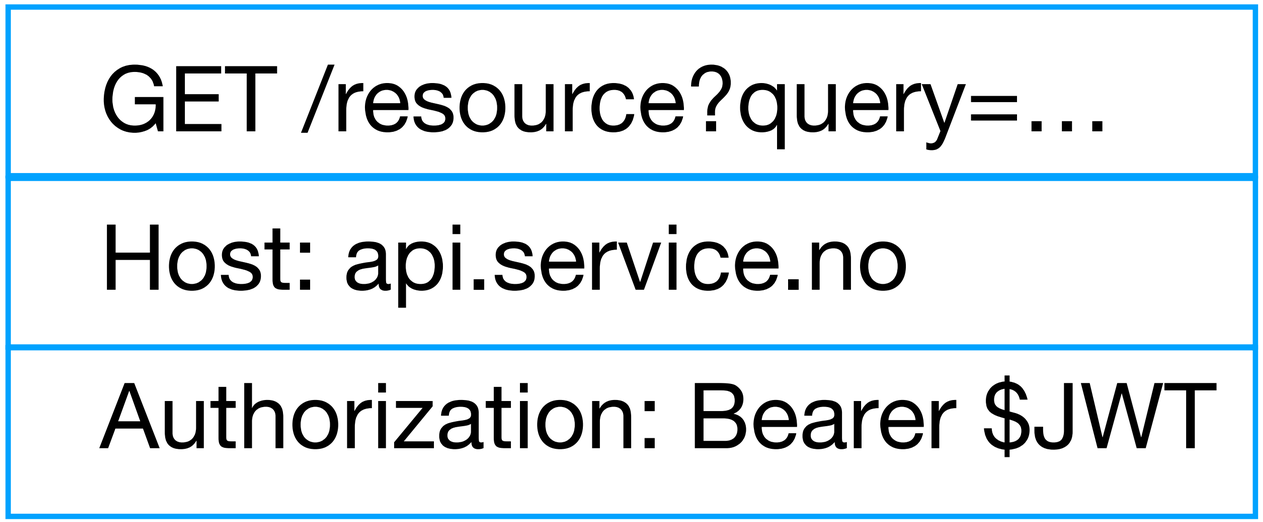
Authorization: Bearer $JWT

Operation, Resource

API

# HTTP Request

**rfc2616#5, rfc9113#4, rfc9114#5**



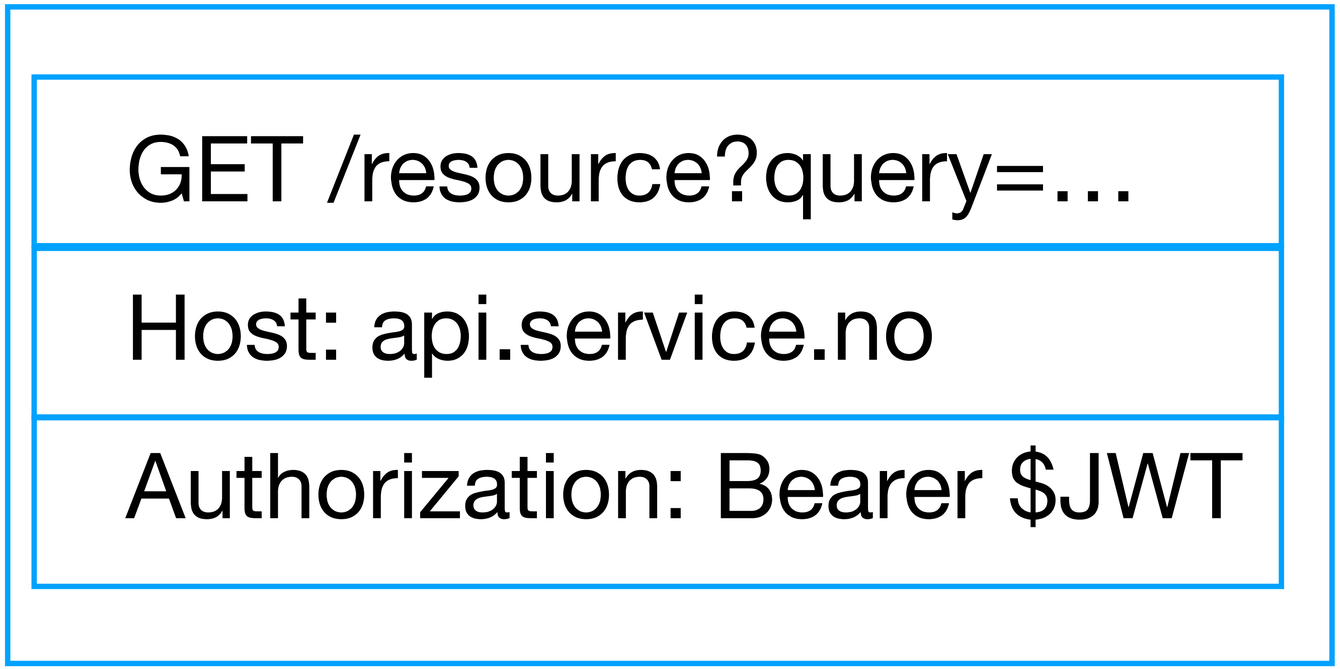

Operation, Resource

API

Claims
(about the user, client, and authentication event)

# HTTP Request

**rfc2616#5, rfc9113#4, rfc9114#5**



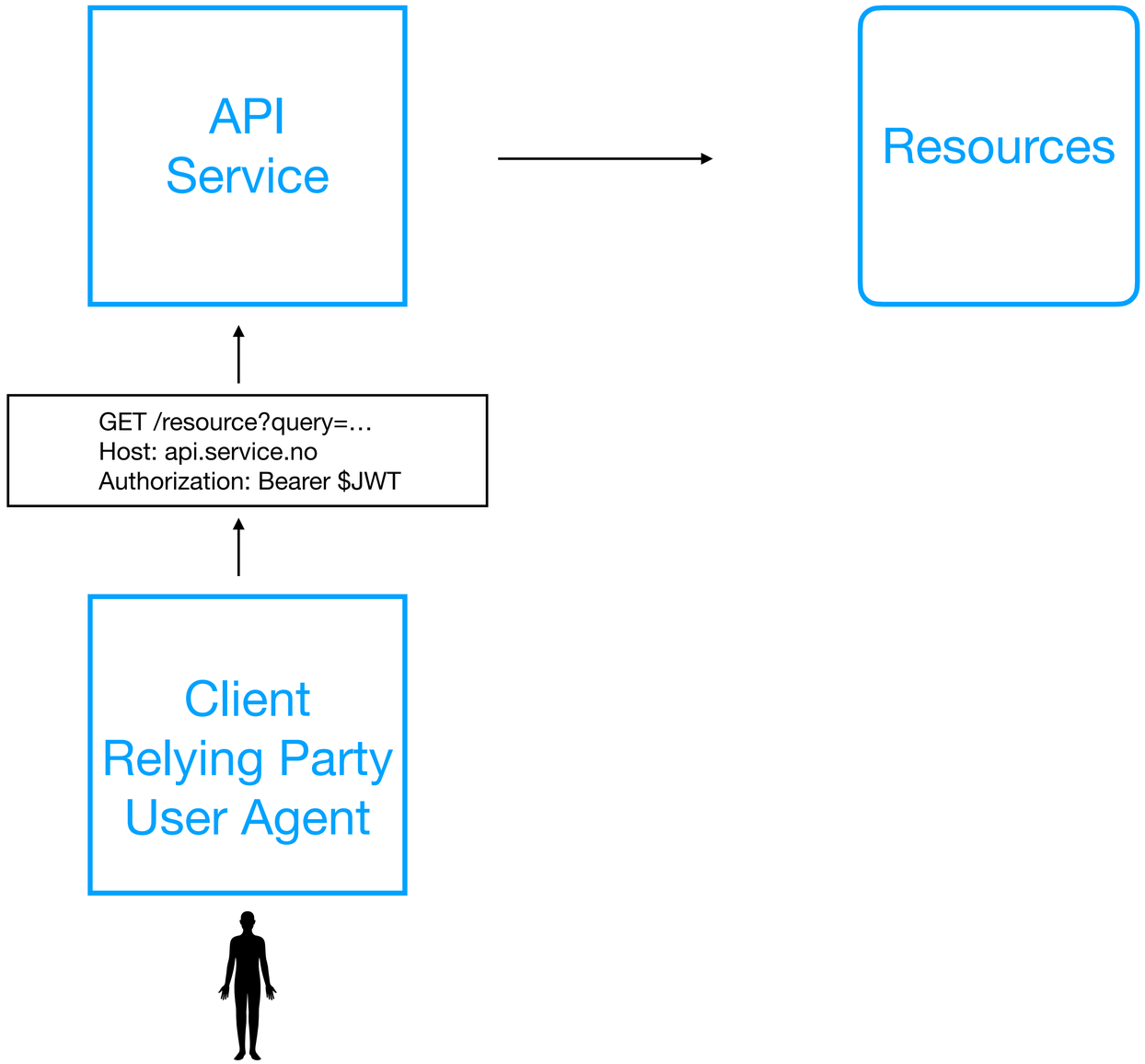

Operation, Resource

API

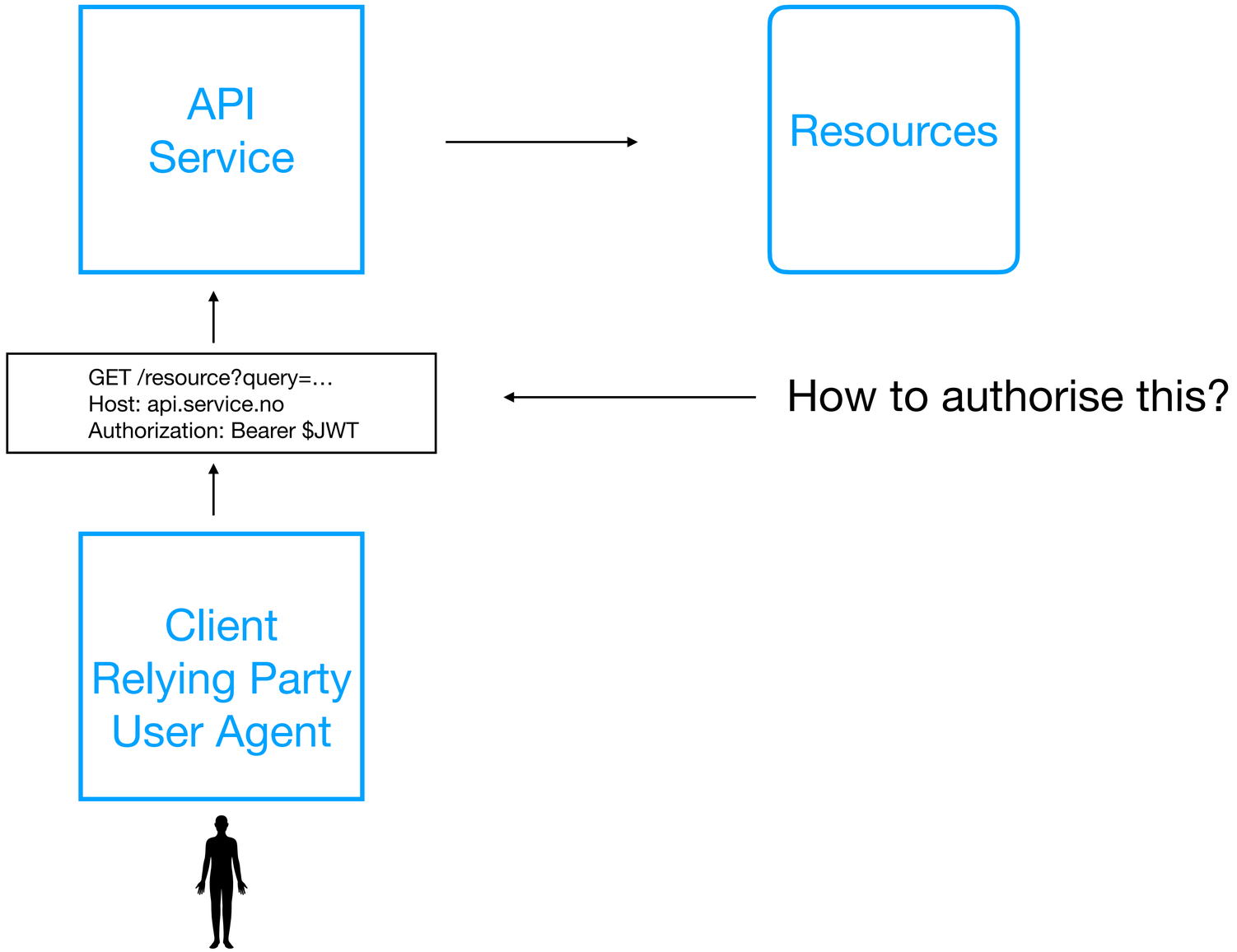
Claims
(about the user, client, and authentication event)

User Agent / Client

# Players


API
Service
Resources
GET /resource?query=…
Host: api.service.no
Authorization: Bearer $JWT
Client
Relying Party
User Agent

# Problem


API
Service
Resources
GET /resource?query=…
Host: api.service.no
Authorization: Bearer $JWT
How to authorise this?
Client
Relying Party
User Agent

# Protocols

API

Resources

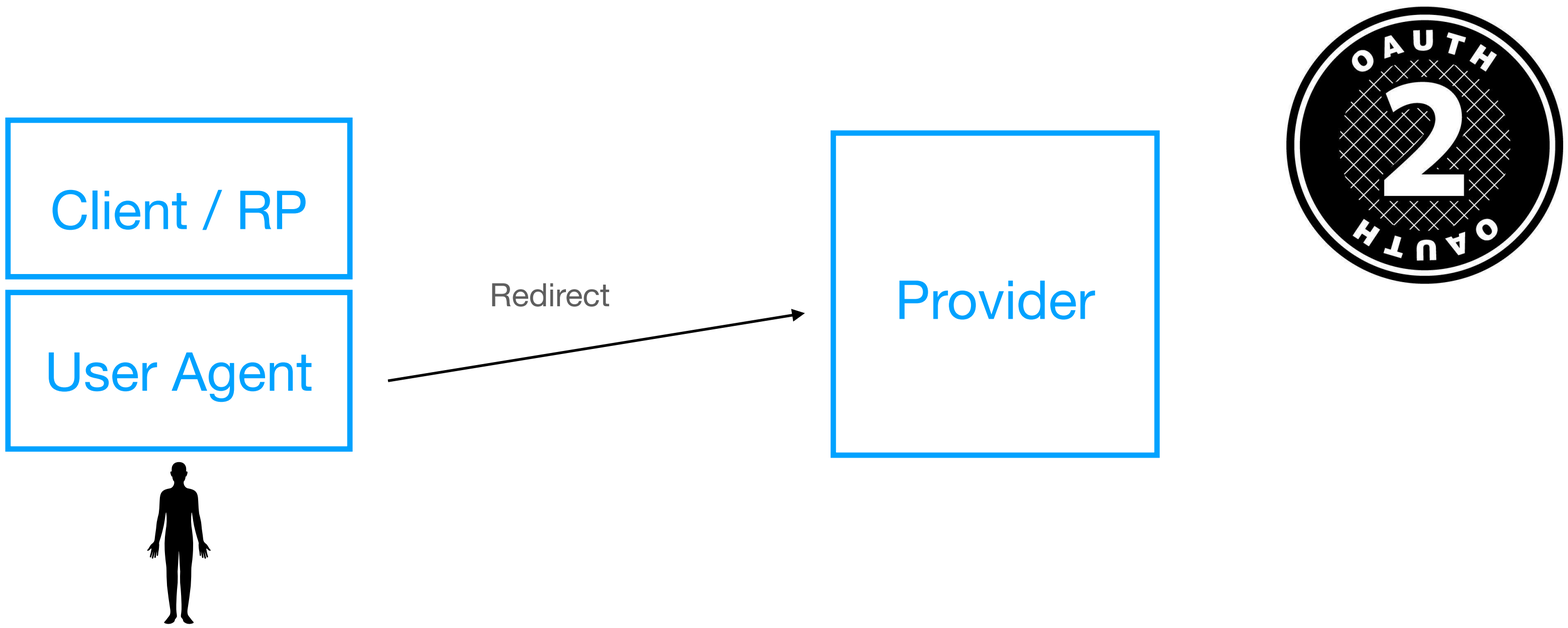
```
GET /resource?query=…
Host: api.service.no
Authorization: Bearer $JWT
```

Client / RP

Provider

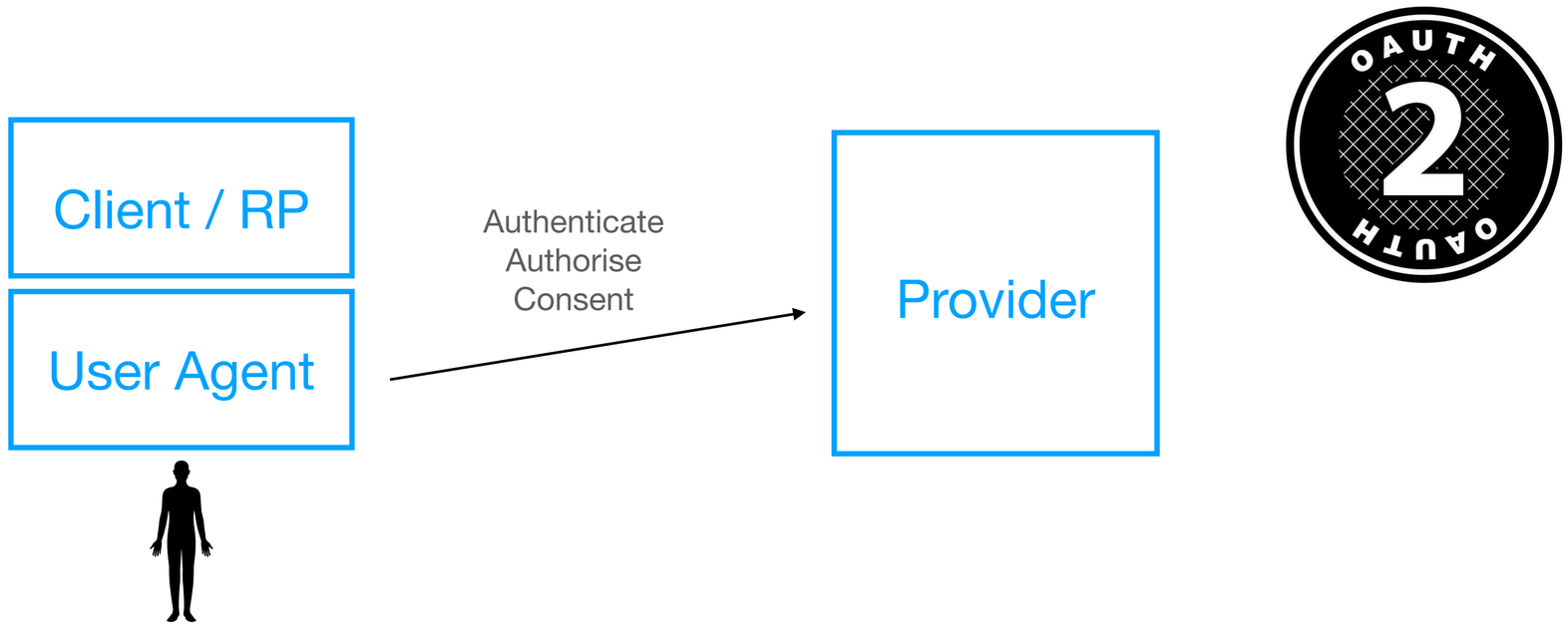

# OAuth2.0

## Example protocol: Code flow



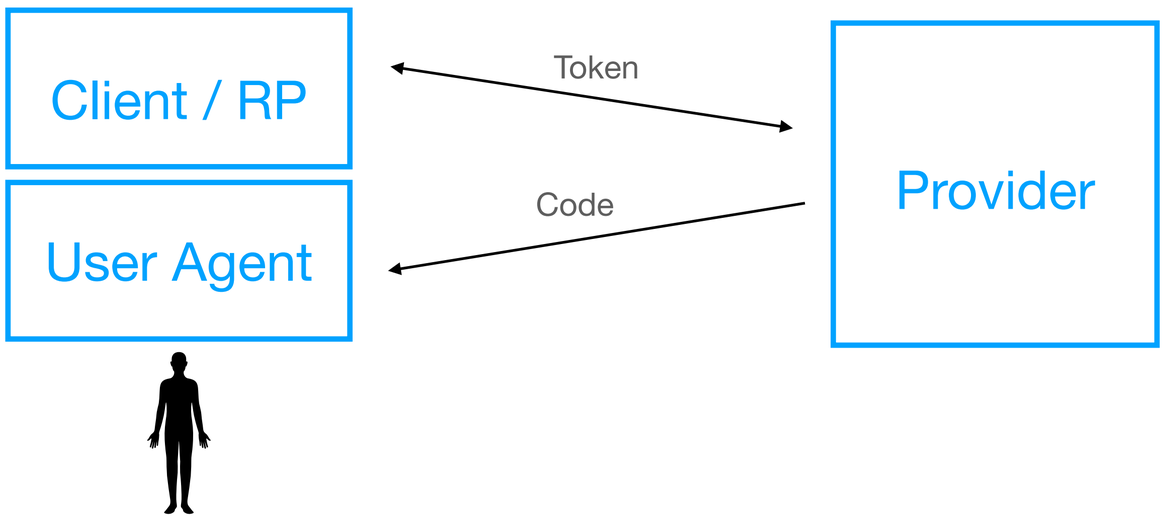
# OAuth2.0

## Example protocol: Code flow



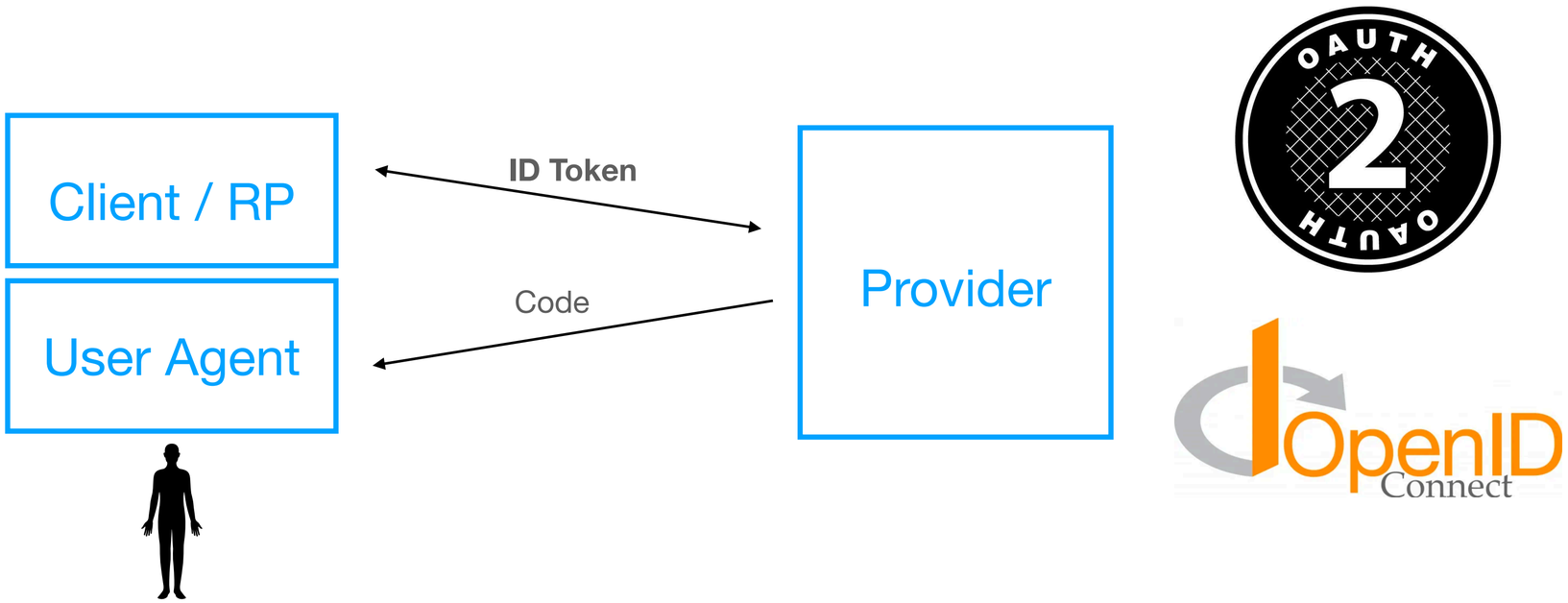
# OAuth2.0

## Example protocol: Code flow

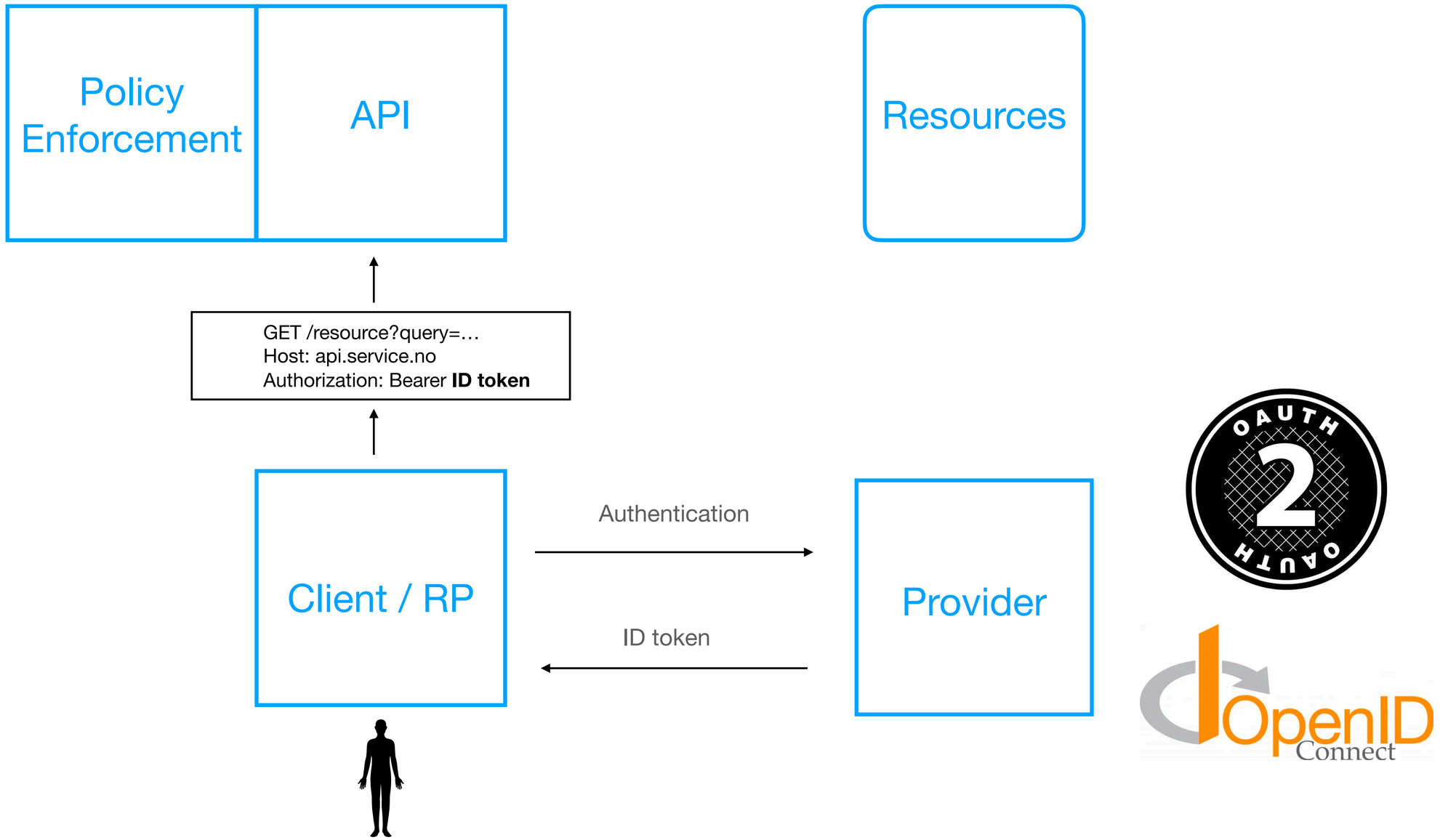



# OpenID Connect

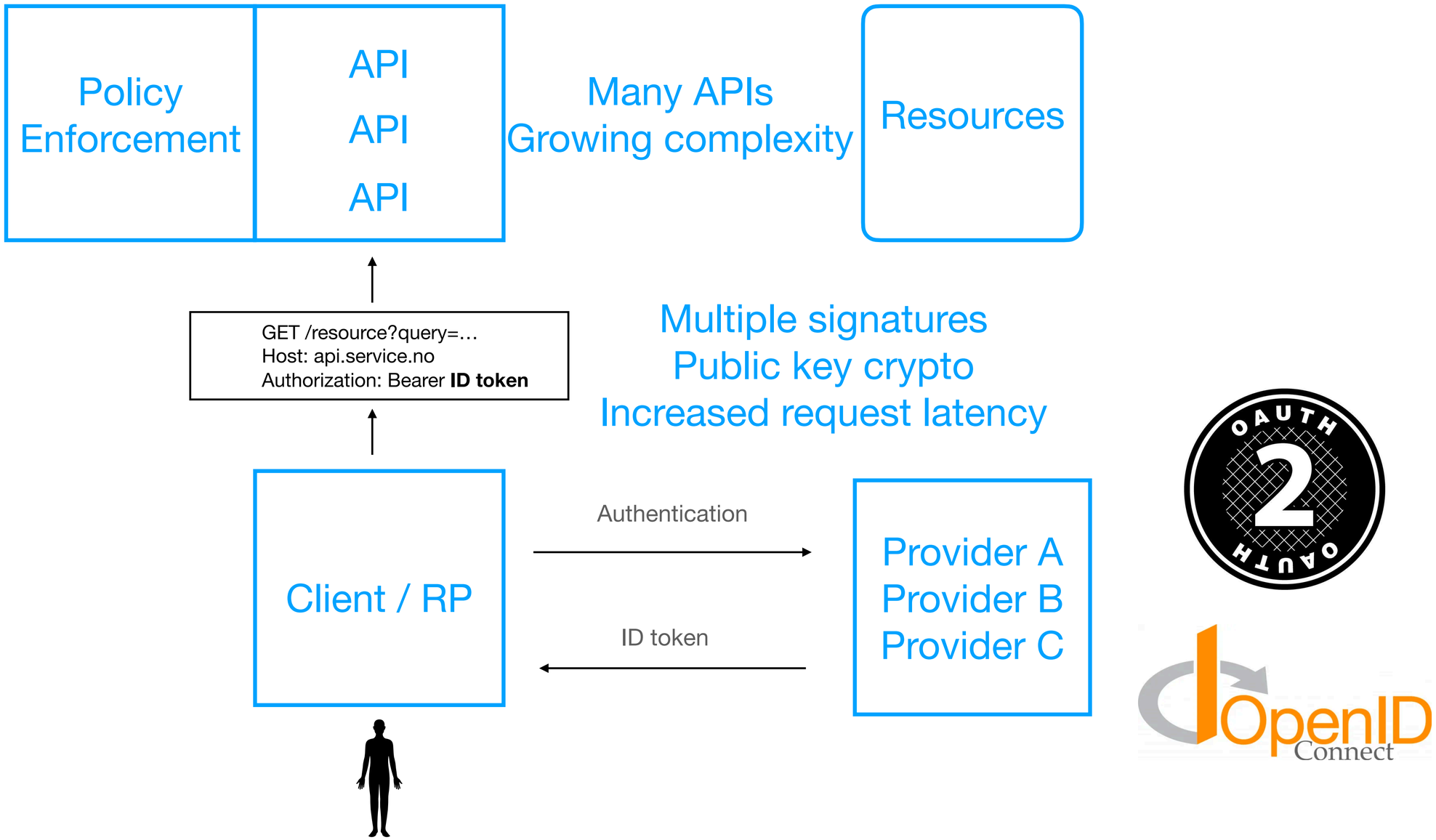
## Adds an identity layer



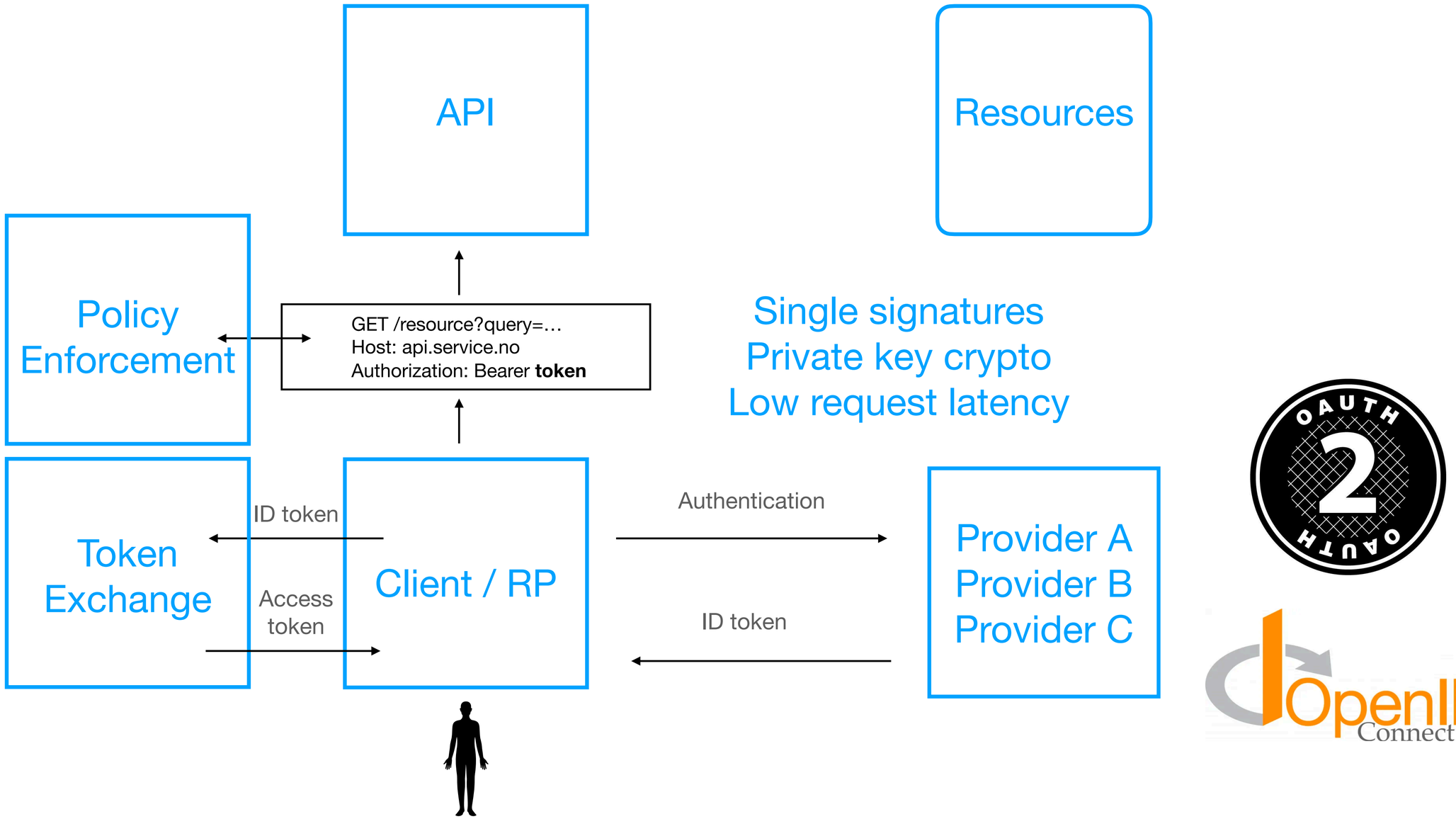
# Solution?


Policy Enforcement
API
Resources
GET /resource?query=…
Host: api.service.no
Authorization: Bearer **ID token**
Client / RP
Authentication
ID token
Provider
OAUTH 2 OAUTH
OpenID Connect

# Complications



# Solution

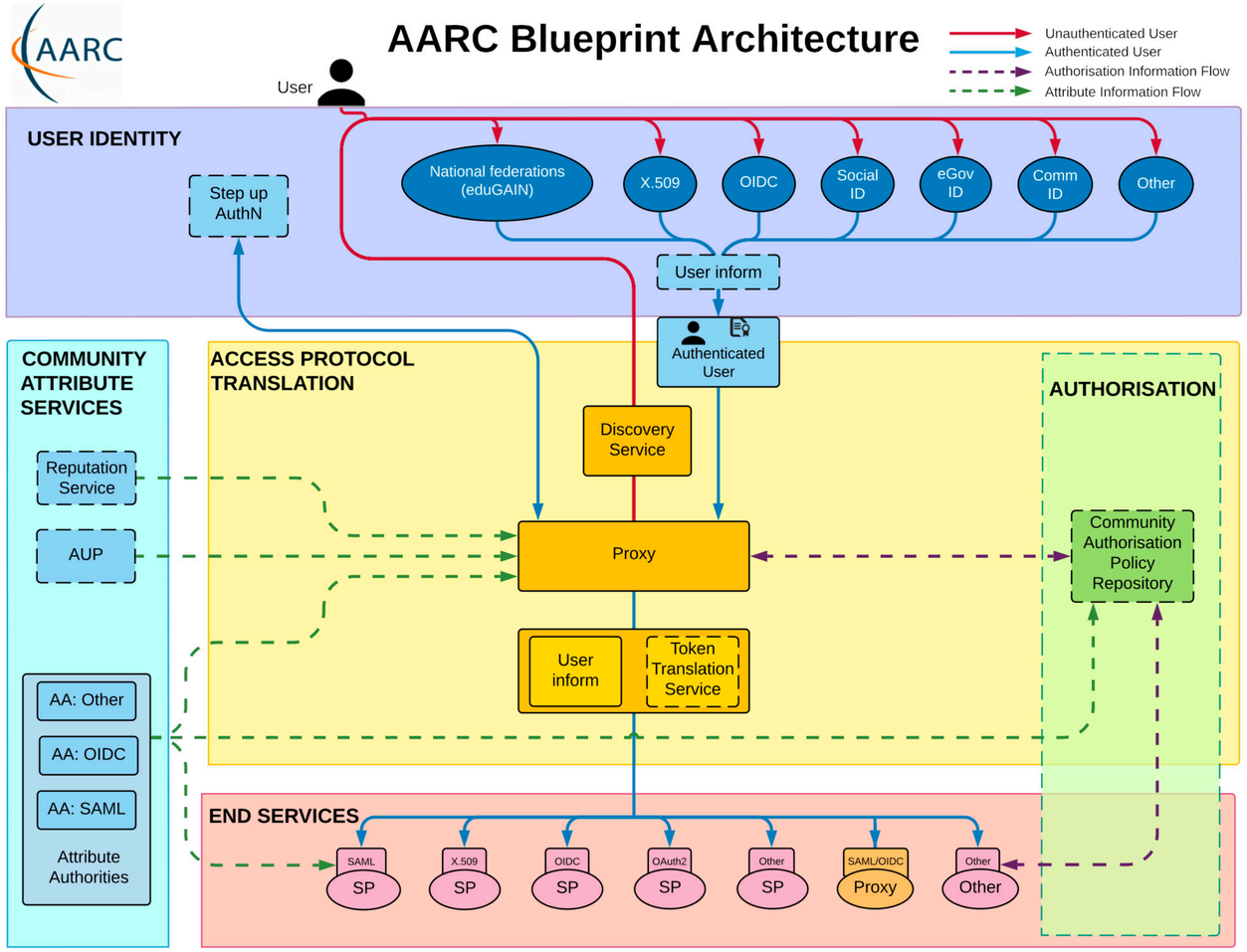

API
Resources
Policy Enforcement
GET /resource?query=…
Host: api.service.no
Authorization: Bearer **token**
Single signatures
Private key crypto
Low request latency
Token Exchange
ID token
Access token
Client / RP
Authentication
ID token
Provider A
Provider B
Provider C


OAUTH
2
OAUTH


OpenID
Connect

# https://aarc-project.eu/

## Authentication and Authorisation for Research Collaborations



# Token exchange

- Exchange ID tokens for JWT access tokens
  - Allows using symmetric crypto for token signatures, HMAC-SHA256
  - Fast verification, low latency authorisation
- API access loosely coupled to 3rd-party OIDC/OAuth providers
- Take control over authorisation semantics

# HTTP Request

**How to perform authorisation?**

GET /resource?query=…

Host: api.service.no

Authorization: Bearer **JWT (exchanged for ID token)**

# OIDC Token response


```
{
    "access_token": "eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiI5NDRiMmQyNS03ODcxLTRj
    NjAtOTFjZi1kNDRkMzhlOTE4YmQiLCJleHAiOjE2MTMwMzI5NzcsImlkIjoicDExLWxlb2
    5jZCJ9.9V4z7GHI27x3avyGtcrxCTpRthHtZgZKhfDKFc-4u4Y",
    "token_type": "Bearer",
    "refresh_token": "eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiI5NDRiMmQyNS03ODcxLTR
    jNjAtOTFjZi1kNDRkMzhlOTE4YmQiLCJleHAiOjE2MTMwMzI5NzcsImlkIjoicDExLWxlb
    25jZCJ9.9V4z7GHI27x3avyGtcrxCTpRthHtZgZKhfDKFc-4u4Y",
    "expires_in": 30,
    "id_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IkZoSFBTNm90NTNmdnZIM3QtSlV3V
    m5hMmVzczNYcUx2dkNuQWl1cWp4dm8ifQ.eyJhY3IiOiJsZXZlbDMiLCJhbXIiOiJUU0Qt
    QXV0aCIsImF1ZCI6Ijk0NGIyZDI1LTc4NzEtNGM2MC05MWNmLWQ0NGQzOGU5MThiZCIsIm
    F1dGhfdGltZSI6MTYxMjk1NDA3OSwiZXhwIjoxNjEzMDMyOTc3LCJpYXQiOjE2MTMwMzEx
    NzcsImlzcyI6Imh0dHA6Ly9sb2NhbGhvc3Q6Mjk5OS92MS9hbGwvdHNkLW9pZGMtcHJvdm
    lkZXIiLCJub25jZSI6Im9hQlAzRGI2dFFpTFNWMVRyRE8wUG5KOTdNc2VhOEZVb1Y4UlpM
    TGwteDgiLCJwcm9qIjoicDExIiwic3ViIjoiIiwidXNlciI6InAxMS1sZW9uY2QifQ.Zvs
    UpnABXswcHcrFWLJFfuUF6Pa7-Rl1gsg2ScjhesU9DU6xvUTQdqFyO97DYWpAtkDBL0Cr9
    spVcOueObsFJS5vtvvltwk3ZND0-91V0eDbpgOe8rpPgjf_L7ykbSLjXVyLg2NxvBPu0aS
    lsPW884yyqo37rzg022WNLbqojZUnpZKyP-7_04BFu7zuMd6FONHjEqZFtzOzJFb0gt_U9
    q_AbQ5WBb3yGmJwNg7NFSkmPBK8OUOomcym90TcFTXvthUzuRr4gooppAC2q48LV_Xr1dR
    vpqJTpA_jjX4BzVA1qFsharmMDexjyx_8oJdfFCm8X4Lqnocyk00K3jnD3w"
}
```

# ID token


```
{
  "sub": "226d4cb4-5e95-40a2-855d-583e5a8930a5",
  "email": "lol@cat.meaow",
  "name": "Lol Cat",
  "aud": "944b2d25-7871-4c60-91cf-d44d38e918bd",
  "auth_time": 1612954079,
  "acr": "level3",
  "amr": ["password", "totp"],
  "iss": "https://oidc.tsd.usit.no/tsd-oidc-provider",
  "iat": 1613031177,
  "exp": 1613032977,
  "nonce": "oaBP3Db6tQiLSV1TrDO0PnJ97Msea8FUoV8RZLLl-x8"
}
```

# ID token

User


```
{
  "sub": "226d4cb4-5e95-40a2-855d-583e5a8930a5",
  "email": "lol@cat.meaow",
  "name": "Lol Cat",
  "aud": "944b2d25-7871-4c60-91cf-d44d38e918bd",
  "auth_time": 1612954079,
  "acr": "level3",
  "amr": ["password", "totp"],
  "iss": "https://oidc.tsd.usit.no/tsd-oidc-provider",
  "iat": 1613031177,
  "exp": 1613032977,
  "nonce": "oaBP3Db6tQiLSV1TrDO0PnJ97Msea8FUoV8RZLLl-x8"
}
```

# ID token

User

Client


```
{
  "sub": "226d4cb4-5e95-40a2-855d-583e5a8930a5",
  "email": "lol@cat.meaow",
  "name": "Lol Cat",
  "aud": "944b2d25-7871-4c60-91cf-d44d38e918bd",
  "auth_time": 1612954079,
  "acr": "level3",
  "amr": ["password", "totp"],
  "iss": "https://oidc.tsd.usit.no/tsd-oidc-provider",
  "iat": 1613031177,
  "exp": 1613032977,
  "nonce": "oaBP3Db6tQiLSV1TrDO0PnJ97Msea8FUoV8RZLLl-x8"
}
```

# ID token

User

Client

Authentication Event


```
{
  "sub": "226d4cb4-5e95-40a2-855d-583e5a8930a5",
  "email": "lol@cat.meaow",
  "name": "Lol Cat",
  "aud": "944b2d25-7871-4c60-91cf-d44d38e918bd",
  "auth_time": 1612954079,
  "acr": "level3",
  "amr": ["password", "totp"],
  "iss": "https://oidc.tsd.usit.no/tsd-oidc-provider",
  "iat": 1613031177,
  "exp": 1613032977,
  "nonce": "oaBP3Db6tQiLSV1TrDO0PnJ97Msea8FUoV8RZLLl-x8"
}
```

# ID token

User

Client

Authentication Event

Provider

# ID token

User

Client

Authentication Event

Provider

Token

# Token exchange

## Enrich claims, own the semantics


```
{
  "sub": "226d4cb4-5e95-40a2-855d-583e5a8930a5",
  "email": "lol@cat.meaow",
  "name": "Lol Cat",
  "aud": "944b2d25-7871-4c60-91cf-d44d38e918bd",
  "auth_time": 1612954079,
  "acr": "level3",
  "amr": ["password", "totp"],
  "iss": "https://oidc.tsd.usit.no/tsd-oidc-provider",
  "iat": 1613031177,
  "exp": 1613032977,
  "nonce": "oaBP3Db6tQiLSV1TrDO0PnJ97Msea8FUoV8RZLLl-x8"
}
```



```
{
  "sub": "226d4cb4-5e95-40a2-855d-583e5a8930a5",
  "email": "lol@cat.meaow",
  "name": "Lol Cat",
  "aud": "944b2d25-7871-4c60-91cf-d44d38e918bd",
  "auth_time": 1612954079,
  "acr": "level3",
  "amr": ["password", "totp"],
  "iss": "https://api.tsd.usit.no",
  "iat": 1613031177,
  "exp": 1613032977,
  "nonce": "oaBP3Db6tQiLSV1TrDO0PnJ97Msea8FUoV8RZLLl-x8",
  "projects": {
    "p11": {
        "name": "ai-mind",
        "start_date": "2021-01-01",
        "end_date": "2030-01-01",
        "institution": "UiO-OUS"
    },
  },
  "groups": ["p11-admin-group", "p11-ai-group", "p11-data-group"],
  "roles": ["admin", "member"]
}
```

# HTTP Request

## The ingredients for policy enforcement

| |
|---|
| GET /resource?query=… |
| Host: api.service.no |
| Authorization: Bearer $JWT |

Operation, Resource

API

Claims - enriched by own IAM system
(about the user, client, and authentication event)

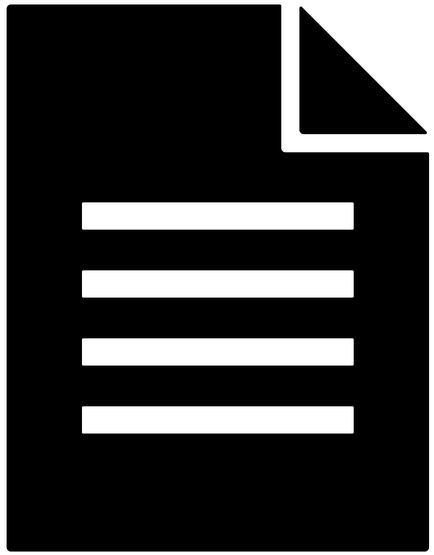
# An example authorisation policy

## Pseudocode

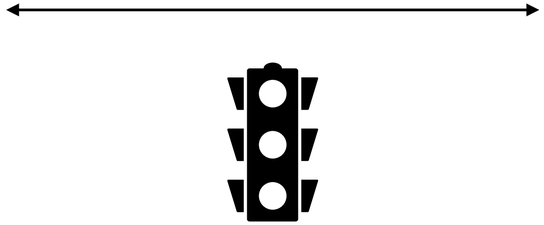
```
Allow
  GET /resources?query=resource-id=([a-zA-Z])
    on Host api.service.com
    if (claims.aud in (registered client-ids) and client-id not expired)
    and int(claims.acr[-1]) >= 3
    and (claims.groups intersects (p[0-9]+-admin-group, p[0-9]+-lol-group)
          or admin in claims.roles)
    and $CURRENT_TIMESTAMP between 2022-01-01 and 2024-01-01
```

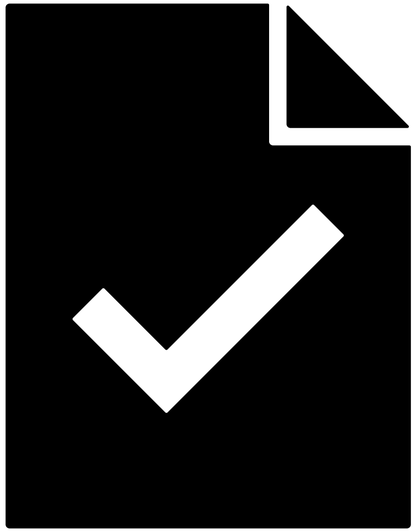
# Policy enforcement

## Conceptually



Given an API call





Describe all possible API calls

Evaluate

Write rules to allow API calls

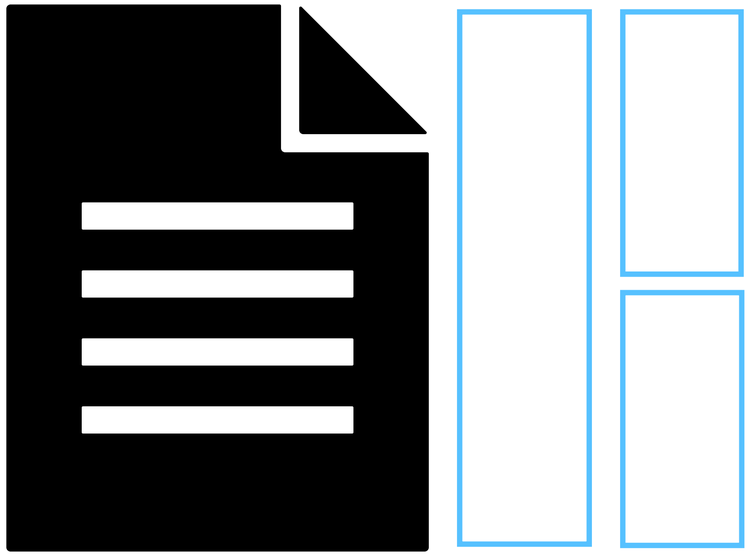
# Policy engines

## Technology choices

- https://www.openpolicyagent.org/
- https://www.cedarpolicy.com/en
- https://zanzibar.academy/
- https://authzed.com/
- https://www.ory.sh/docs/keto

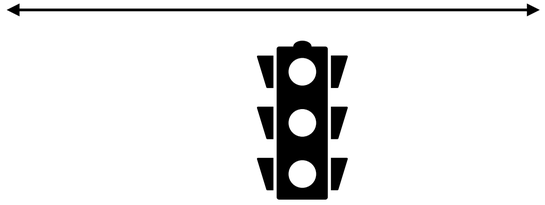
# Limit access per application

## Scopes

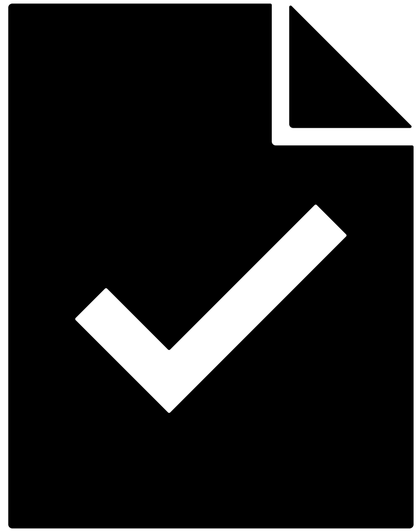


Describe all possible API calls

Given an API call



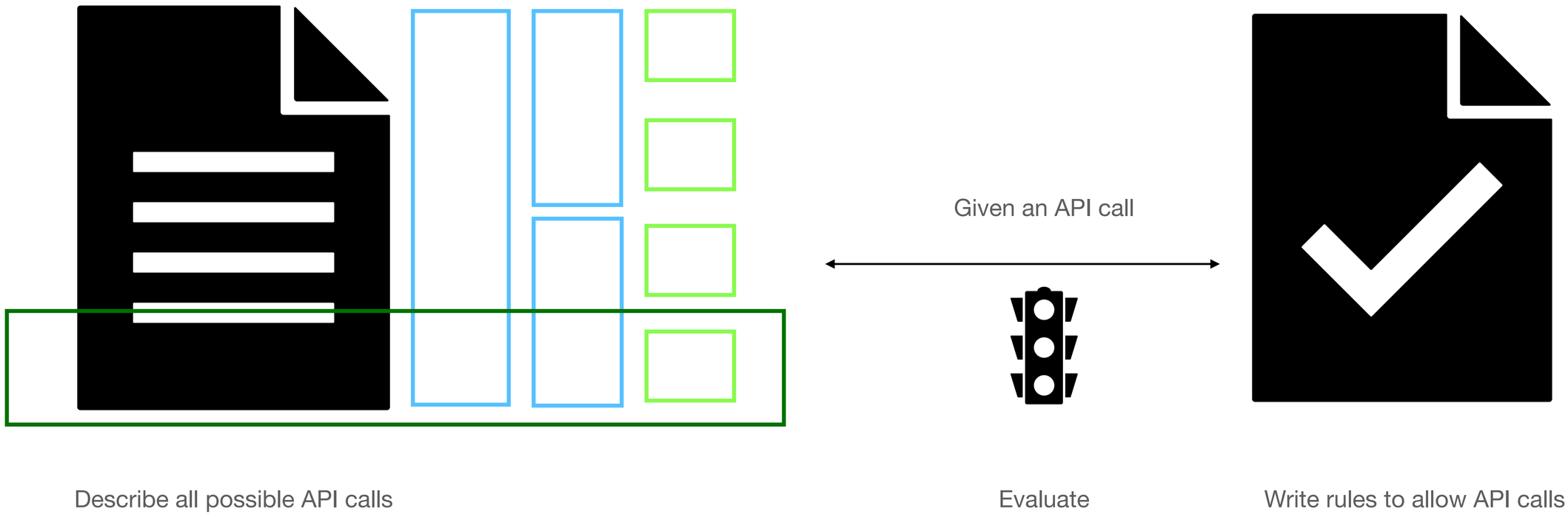
Evaluate



Write rules to allow API calls

# Limit access per application, per user

## Scopes



# An example authorisation policy

## Leverage: claims, and scopes

Allow

GET /resources?query=resource-id=([a-zA-Z])

on Host api.service.com

if (claims.aud in (registered client-ids) and client-id not expired)

and claims.scopes intersects (metadata, audit)

Client

and int(claims.acr[-1]) >= 3

and (claims.groups intersects (p[0-9]+-admin-group, p[0-9]+-lol-group)

or admin in claims.roles)

User

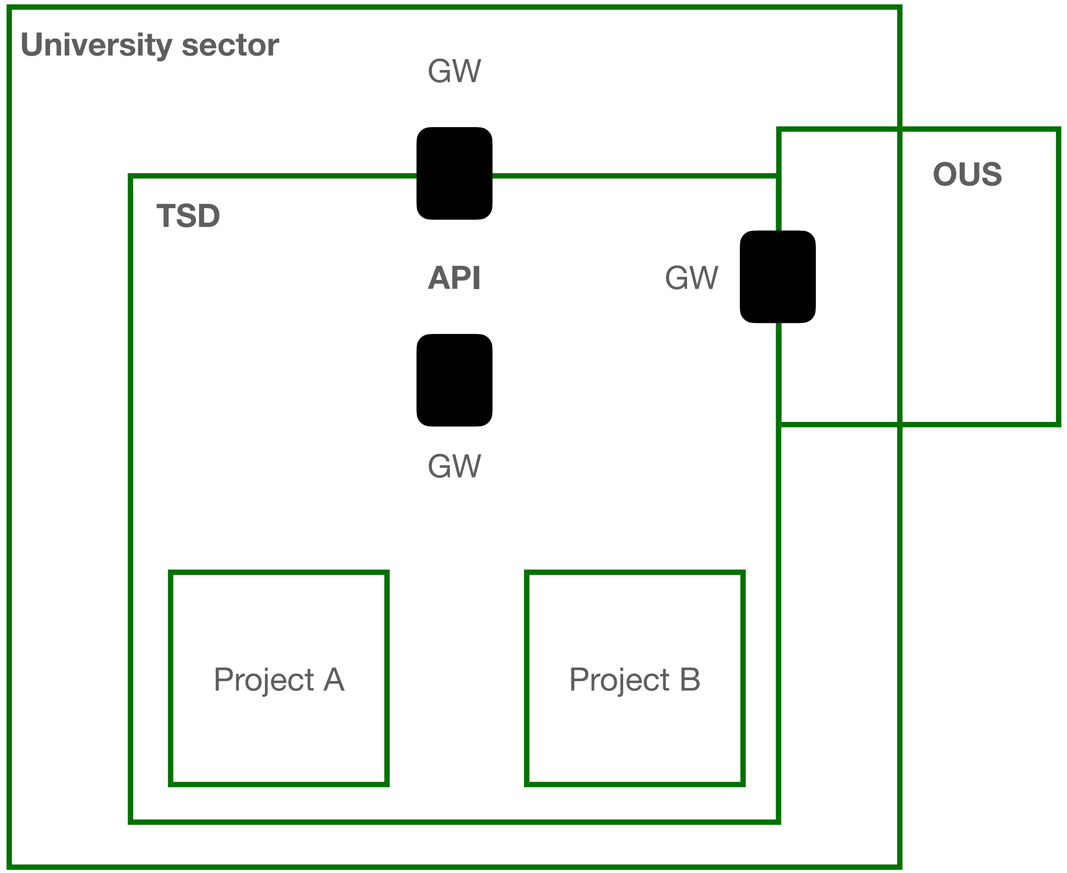
and $CURRENT_TIMESTAMP between 2022-01-01 and 2024-01-01

# Experience at UiO/TSD

- TSD: Secure on-premises cloud for research with sensitive data
- 8PiB data, 2000 research projects, 8000 users, 3000 VMs
- Own IAM: OIDC provider, 2FA, API authorisation
- VmWare remote login
- App development platforms
- HTTP APIs for everything

# The world according to the TSD API

EU/EEA GDPR | non-EU/EEA GDPR equivalent | No GDPR, allies | No GDPR, not allies



# Three API Gateways

## Different trust levels

Allow

GET /resources?query=resource-id=([a-zA-Z])

on Host **api.tsd.usit.no** | **internal.api.tsd.usit.no** | **alt.api.tsd.usit.no**

if (claims.aud in (registered client-ids) and client-id not expired)

and int(claims.acr[-1]) >= 3

and (claims.groups intersects (p[0-9]+-admin-group, p[0-9]+-lol-group)

or admin in claims.roles)

and $CURRENT_TIMESTAMP between 2022-01-01 and 2024-01-01

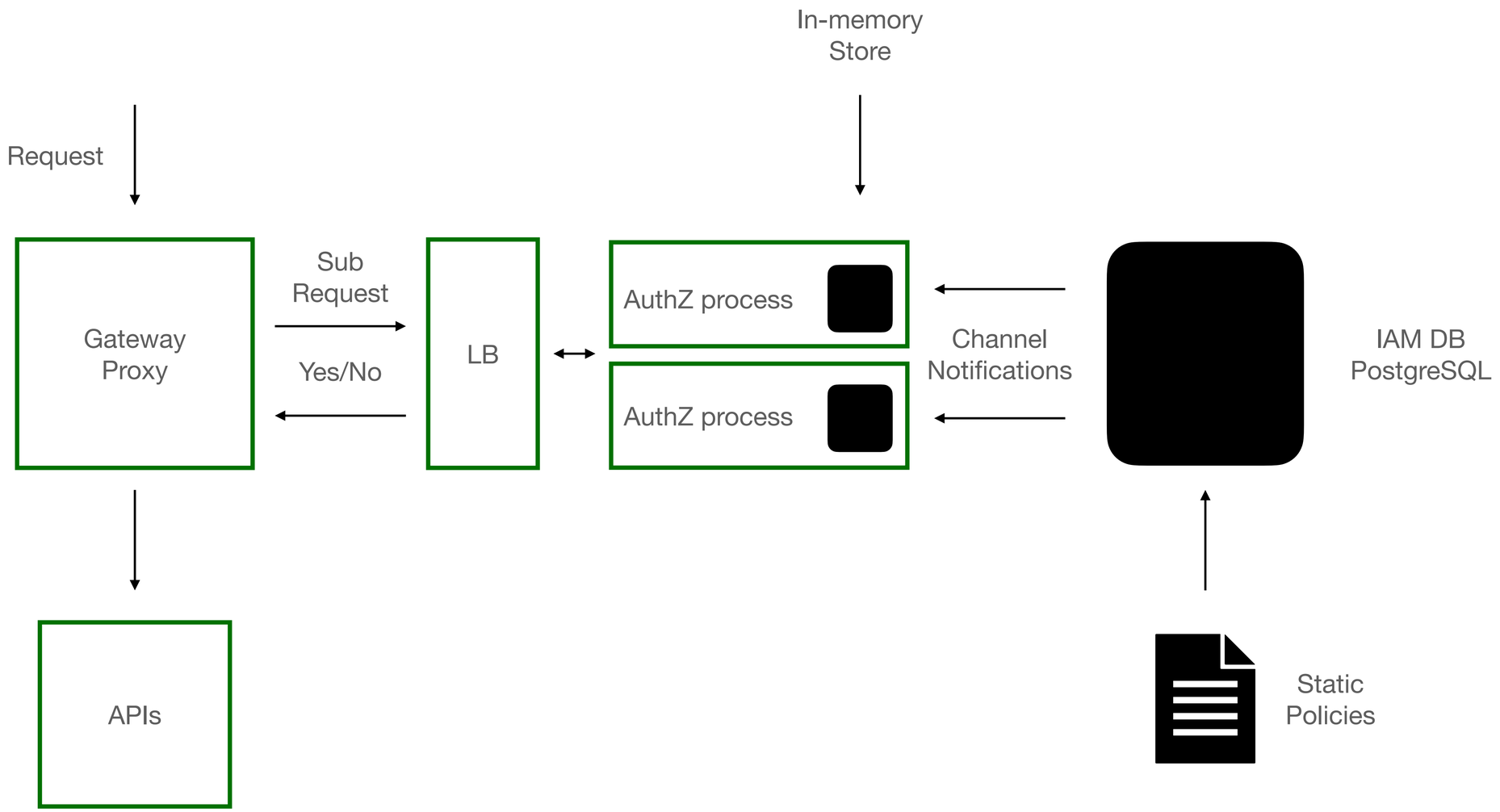
# TSD-IAM

## HTTP authorisation

- Supports four OIDC providers, three 3rd party
- 242 static authorisation policies
- 19 357 programmatic rules - created by applications
- 1ms request latency

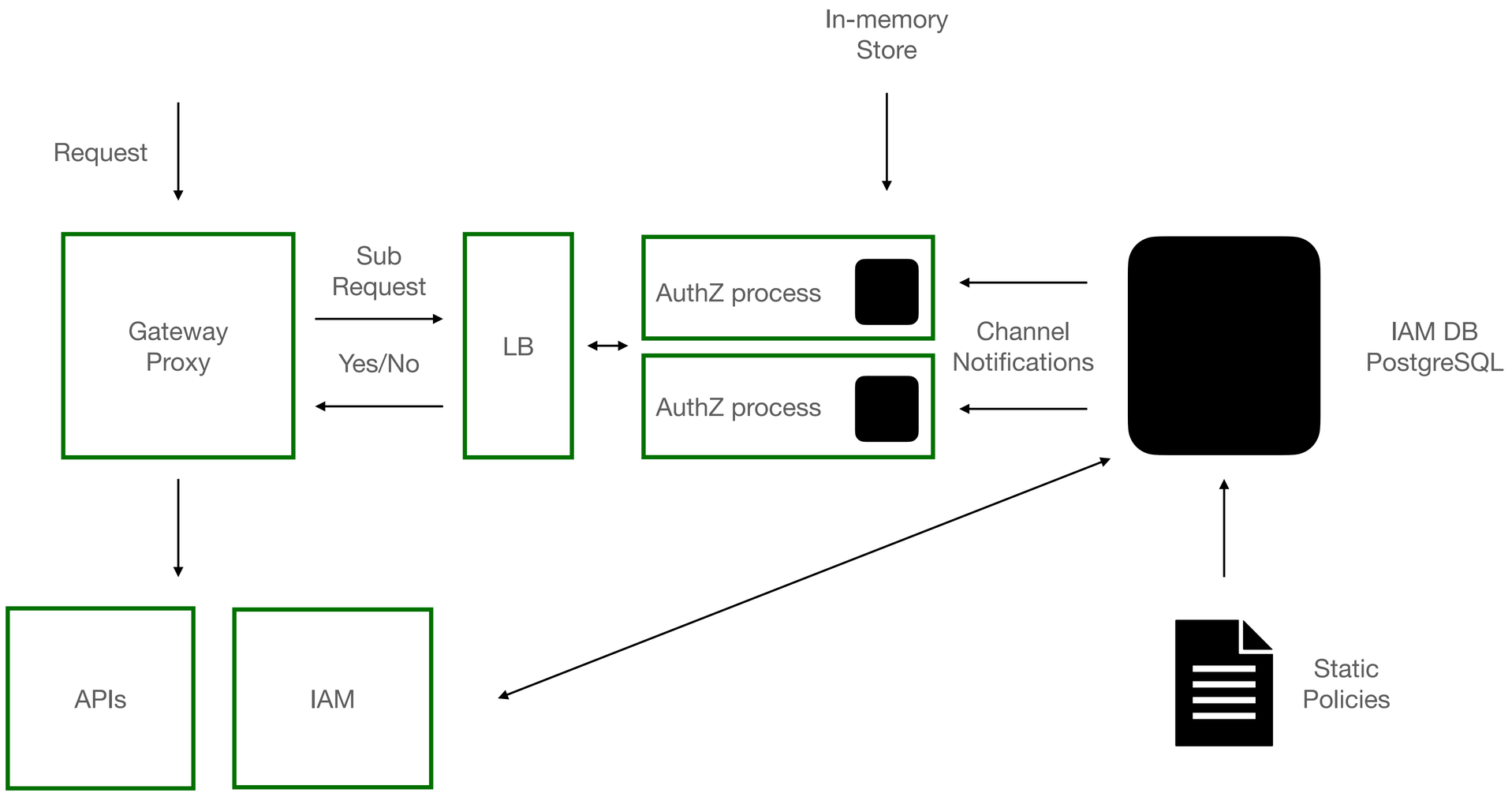
# TSD-IAM

## Implementation



# TSD-IAM

## Implementation



# TSD-IAM

## Implementation

- Open source DB: https://github.com/unioslo/pg-iam

# Recommendations

- Start with high level policies
- Find the relevant geopolitical and institutional boundaries
- Map the relevant OAuth/OpenID Providers
- Use token exchange: enrich claims, add scopes
- Centralise authorisation (most of it)
- Minimise static policies
- Enable programmatic policies

# Thanks

## Questions?

- leoncd@uio.no