20 MAY 2025

DEVOPS DAYS 2025

API 🔑 AuthN / AuhtZ

Demystifying OAuth2, OpenID Connect, and Modern Authentication Techniques

Benjamin Calvet

SysOps Engineer

@l0rd_r4

in/benjamin-calvet

Quentin Rodic

Crazy Developer

@tintin92350

in/quentin-rodic

AuthZ / AuthN ?

- AuthN stand for **Auth**e**N**tication.
- AuthZ stands for **Auth**ori**Z**ation.

AGENDA

On today’s journey

1. Presentation (30 min)
2. Workshop (30-40 min)
3. OpenSpace around Authorization (20-30 min)

Workshop Context

CONTEXT

DevOpsBeerer app

Our Buisness application, fully Open Source :)

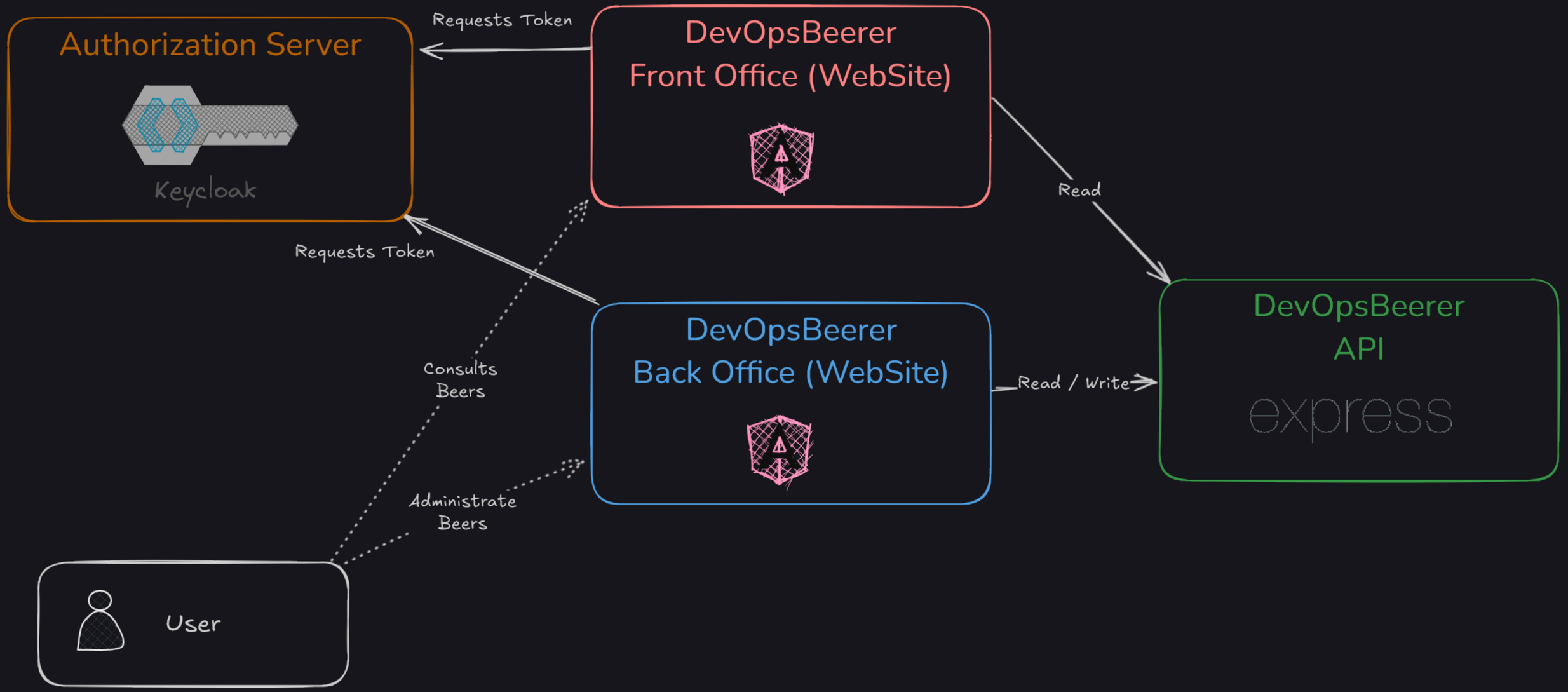
- Users can check Beers served at DevOpsDays.
- Open Source project,
- Secure transactions.

Technologies :

- **Frontend** : Angular
- **Back Office** : Angular
- **Backend** : ExpressJS
- **Authorization Server :** Keycloak

DEVOPSBEERER

DevOpsBeerer Architecture

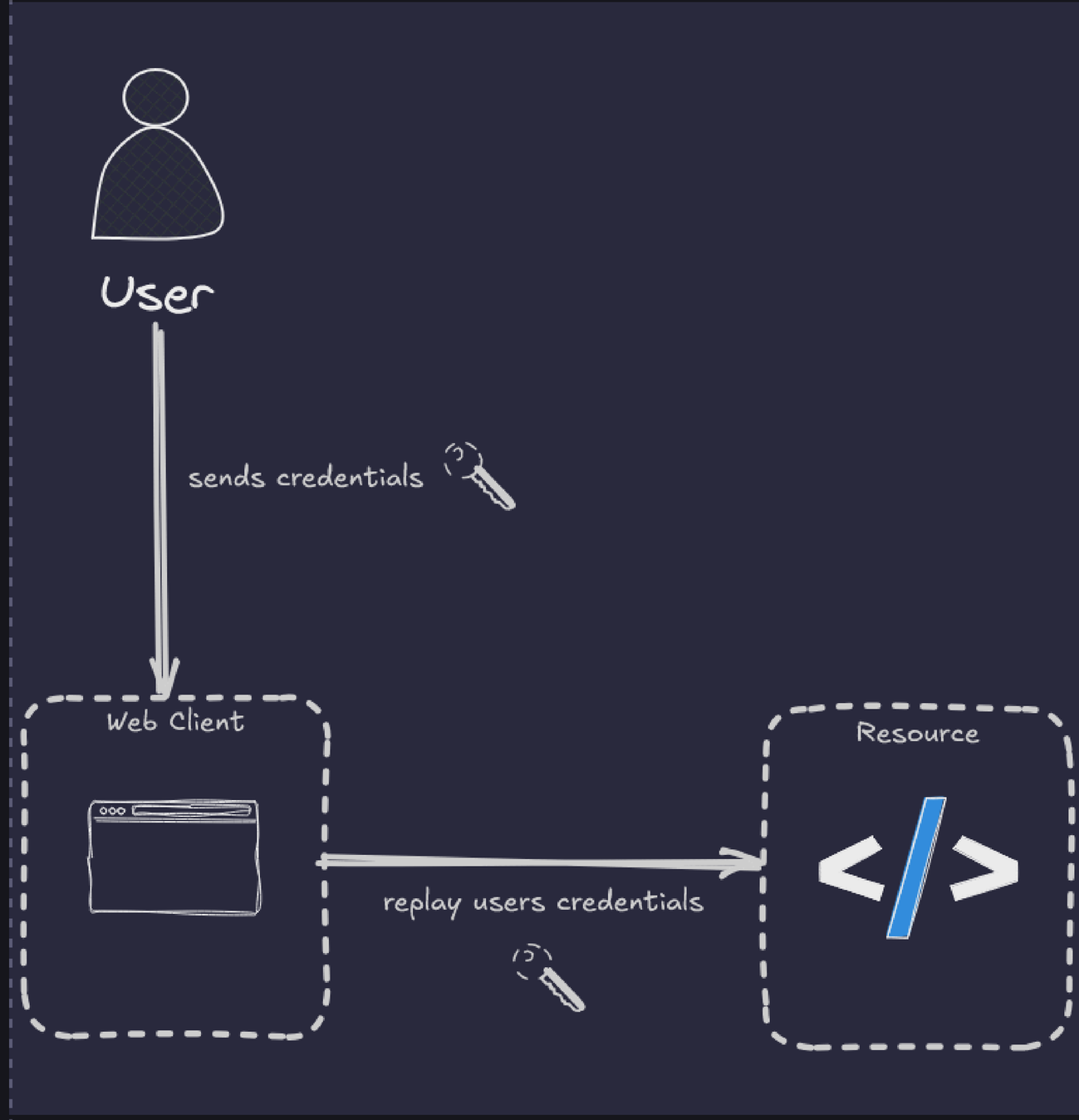


Origins

The bad ol’days

Credentials sharing pattern, could lead to theft..

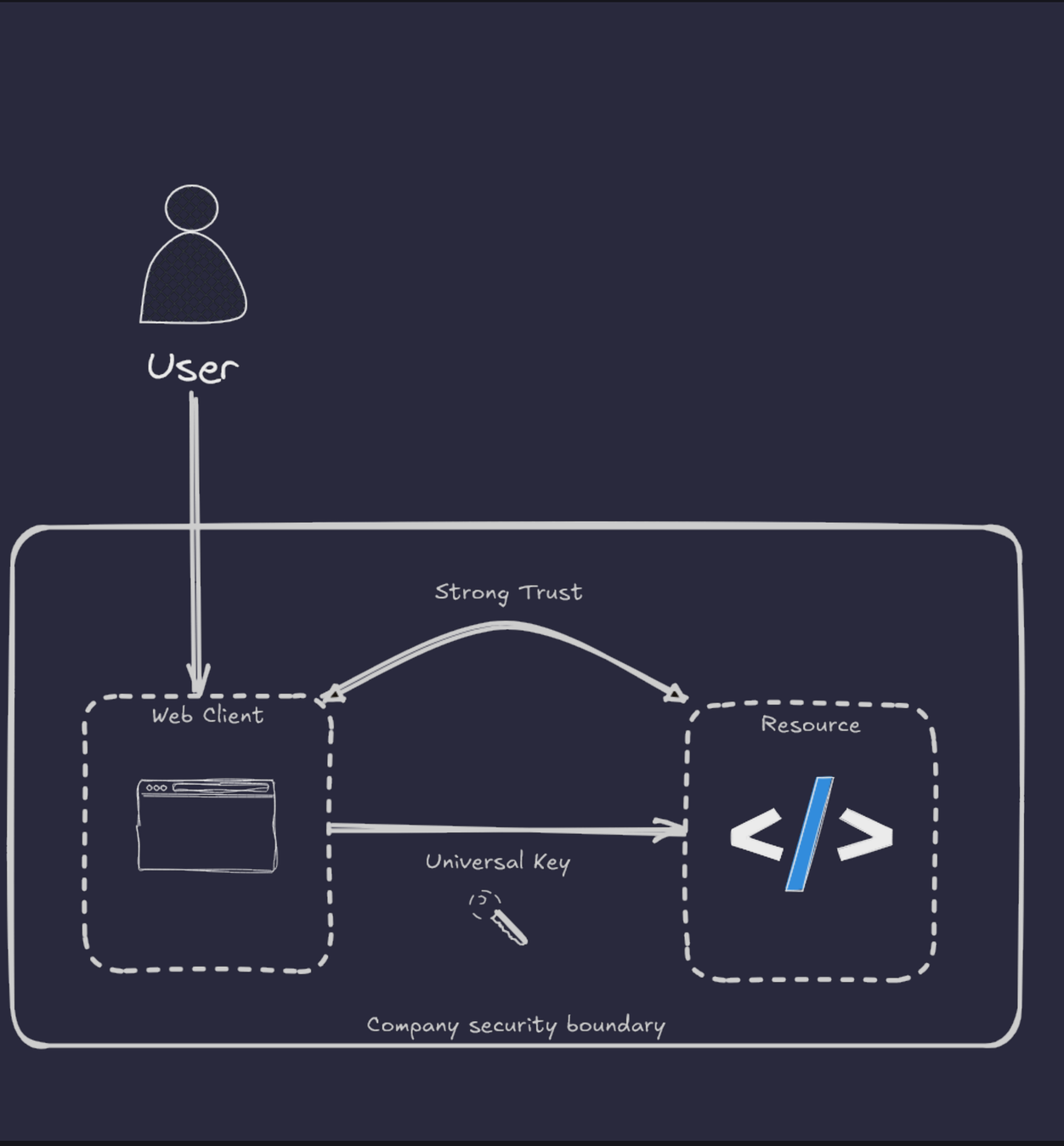
- User impersonnation on third party apps.
- Replayed each time on each app.
- LDAP



Improvment : Developer/API Key

- Other common way, single key to authenticate user
  Could also be a special password only for the client

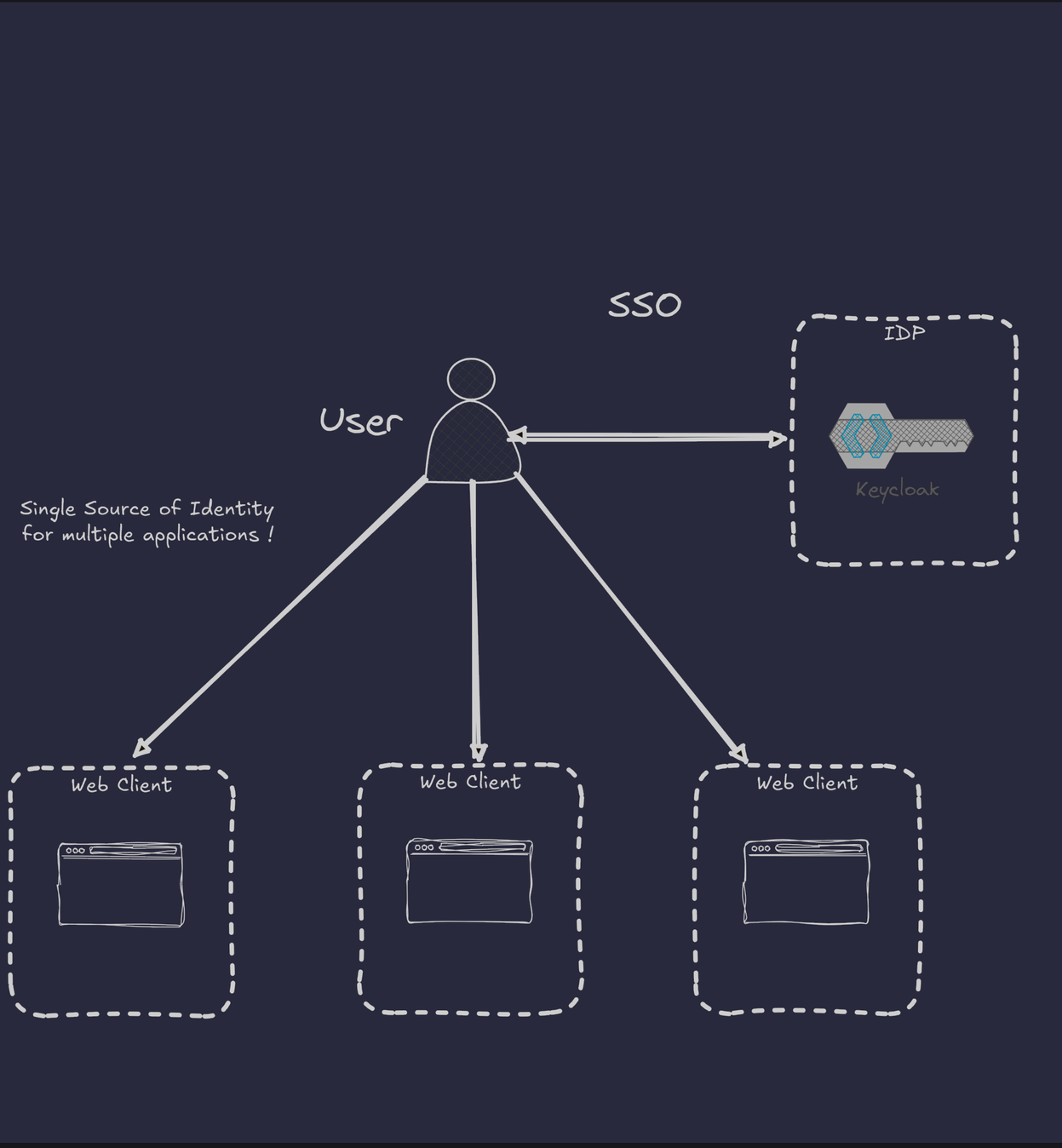
- Impacts all users of the application if credential leaked.



Enterprise Needs
For Security

SSO - Single Sign-On

- Reduce password fatigue, enabling users to enter their credentials only once in a session-time.
- I only have to enter my password once to view my beers the whole day !



FIM - Federated Identity

👉 Beyond SSO !
Collaborate with multiple organizations / tenants with multiple sources for indentities.

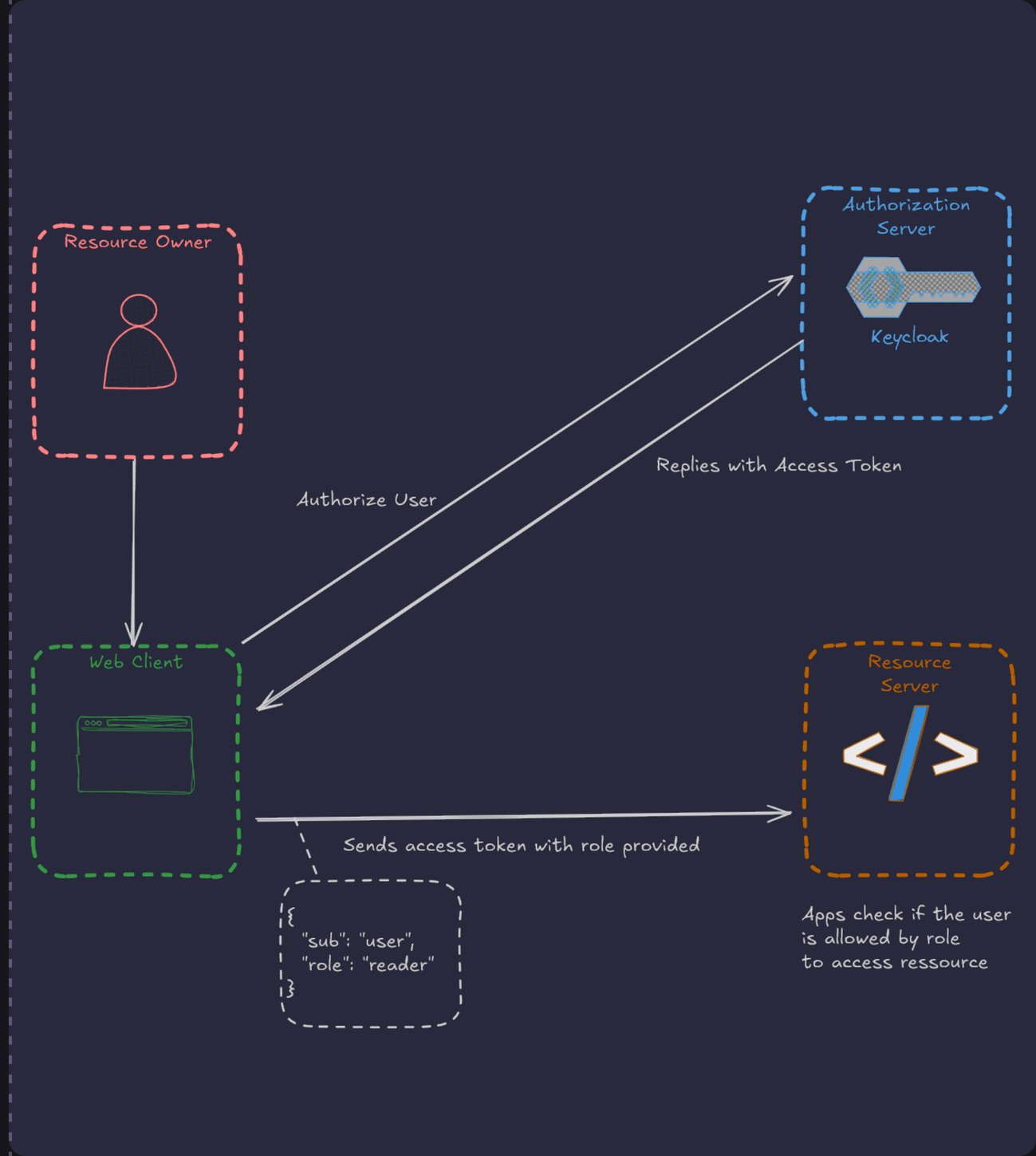
- B2B Collaboration
- Protocols : WS-Trust, WS-Federation

AuthZ : How to authorize ?

- Authentication is the first step of Authorization
- Define an Auhtorization Model :
  - **CGA** : Coarse-Grained Authorization
  - **FGA** : Fined-Grained Authorization
- Pick an Access Control method :
  - **RBAC / ReBAC / ABAC / CBAC / UBAC**
- Key elements of AuthZ:
  - **Subject** : The requester.
  - **Action** : read, write, delete.
  - **Resource**: The targeted object.
  - **Context** : Other variables to be considered

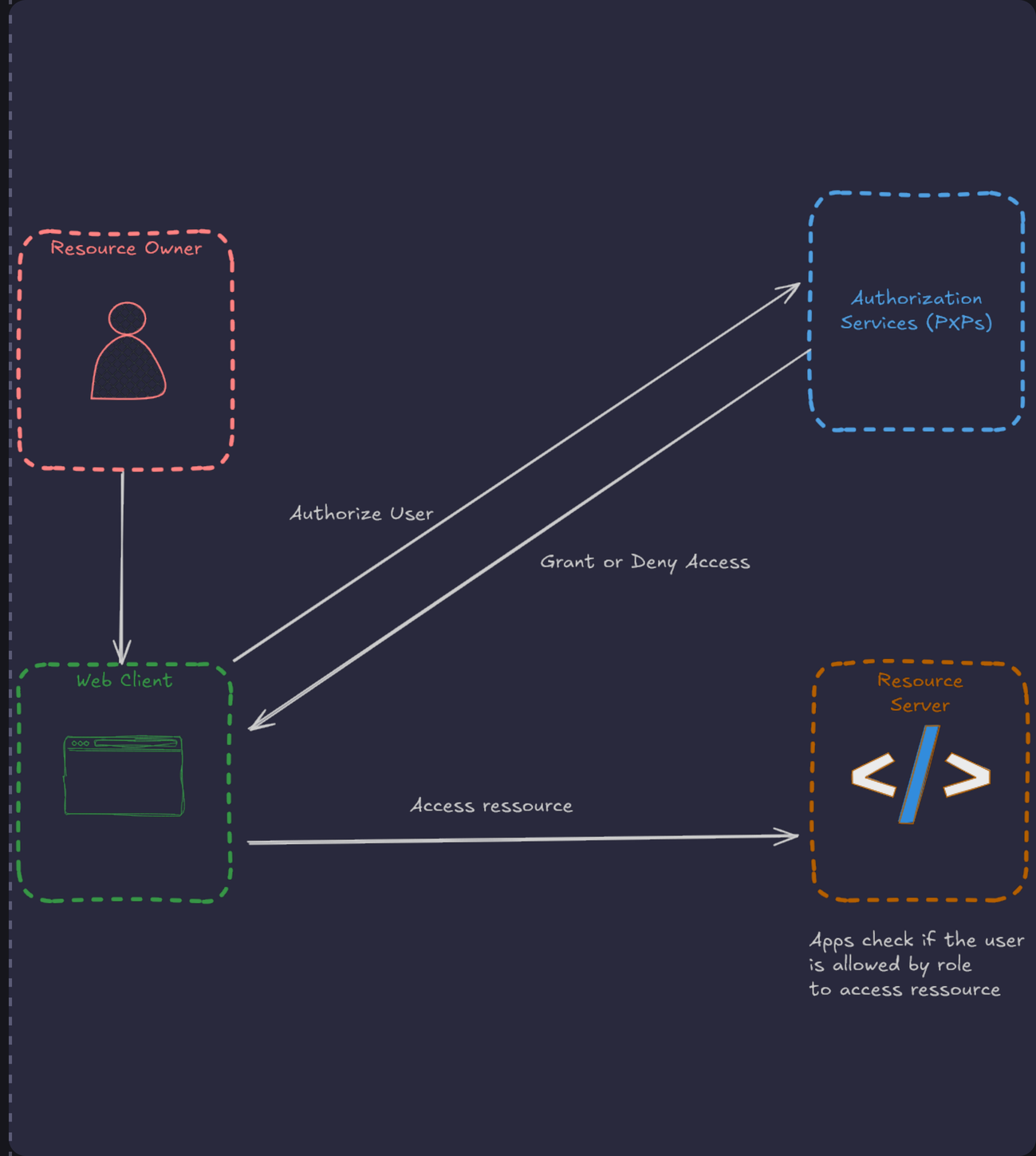
CGA : First Approach

- Roles defined per user ou group of users
- Replicates the organization permissions structure
- Low setup required, easy “getting started”.
- Enough for applications with small buisness logic



FGA : Leap Ahead

- Can combine multiple access control based on policies.
- Rely on authorization services (relational)
- More granular but more complex.
- Useful when application buisness needs are evolving or becoming to complex to manage with roles

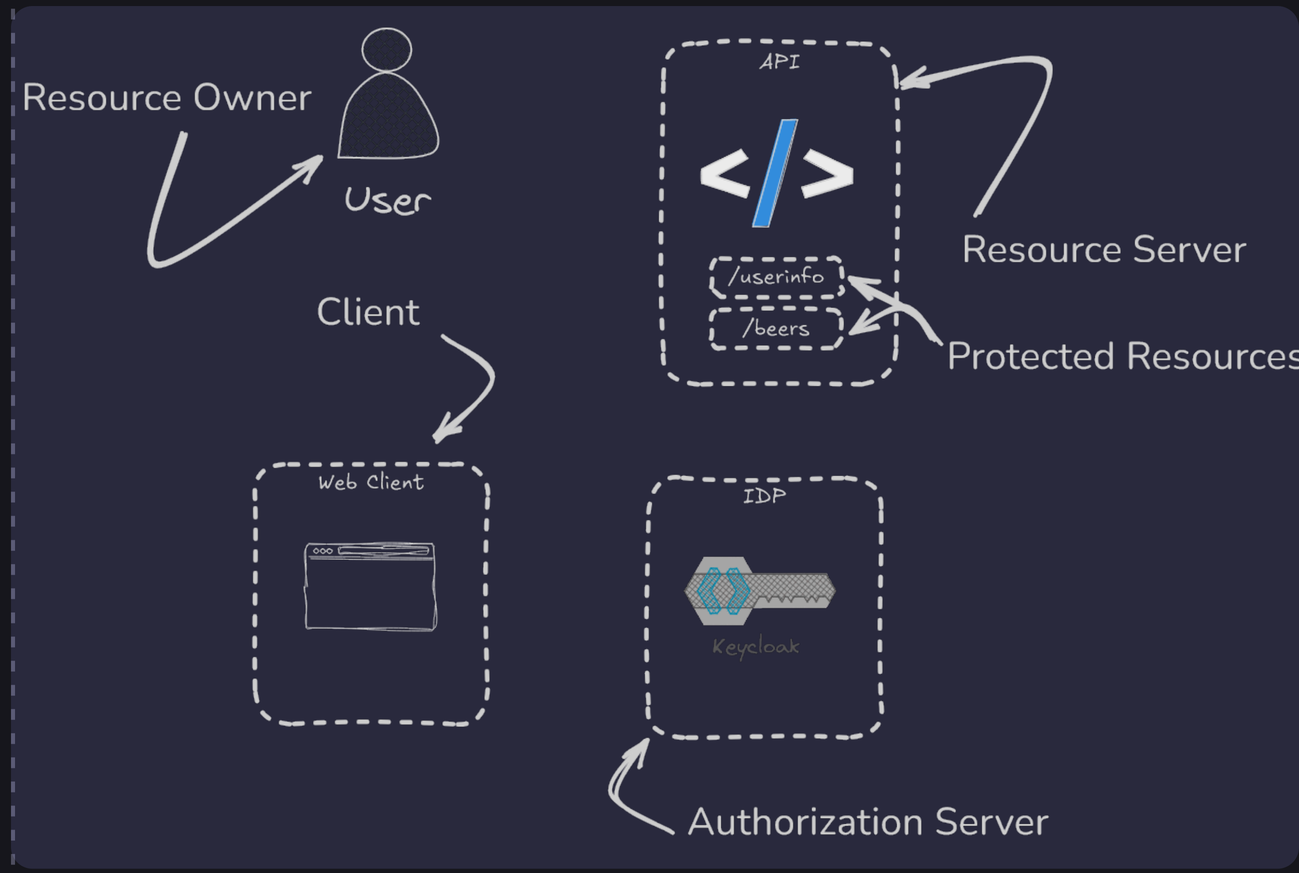


And Now...
OAuth2.0

RFC #6749

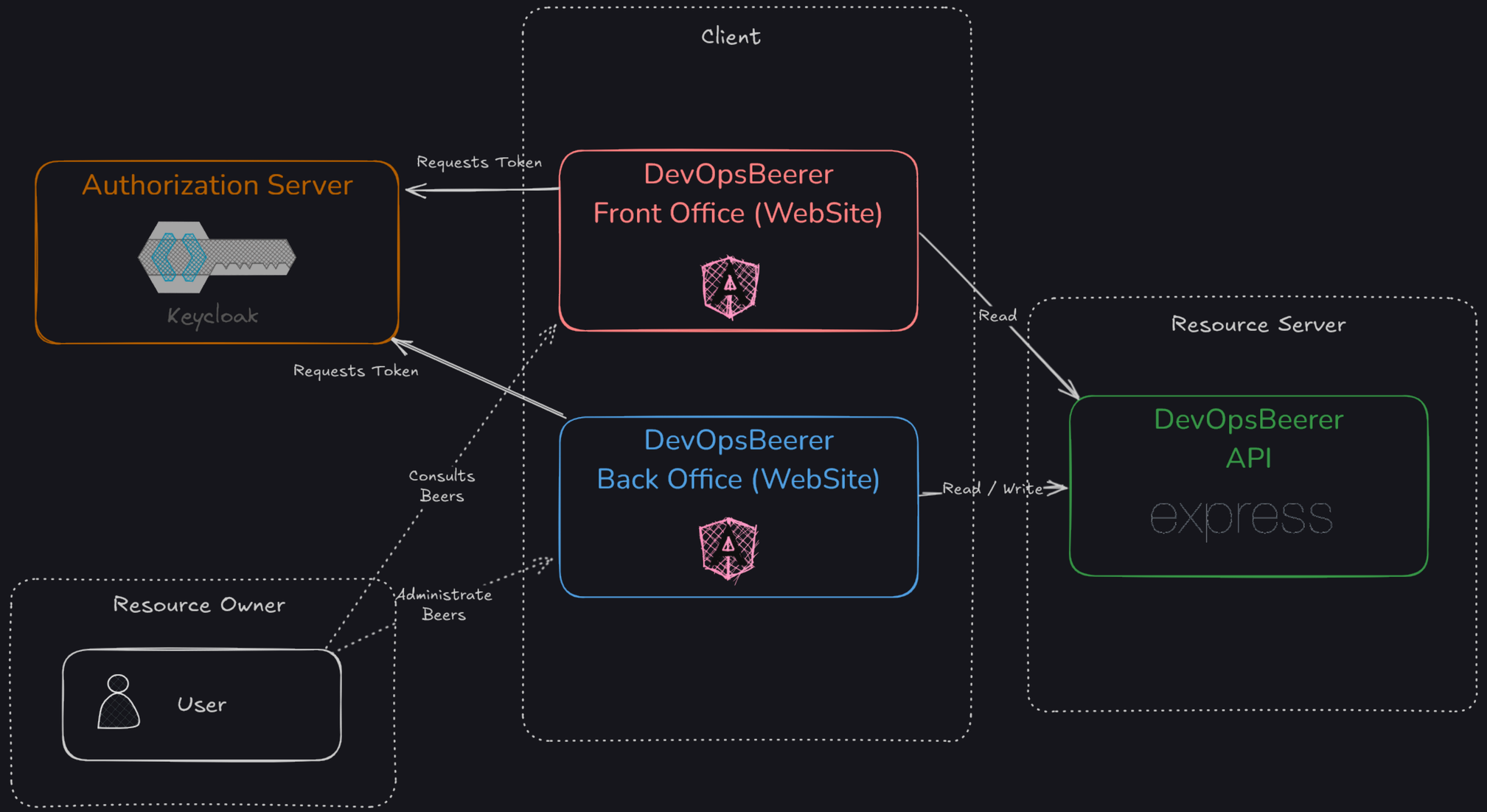
OAuth 2.0 : A Delegation **Protocol**

- Access ressource on user behalf without impersonation.
- Grant third-party apps to acess data upon user consent (TOFU).
- OAuth stands for : **Open Authorization**



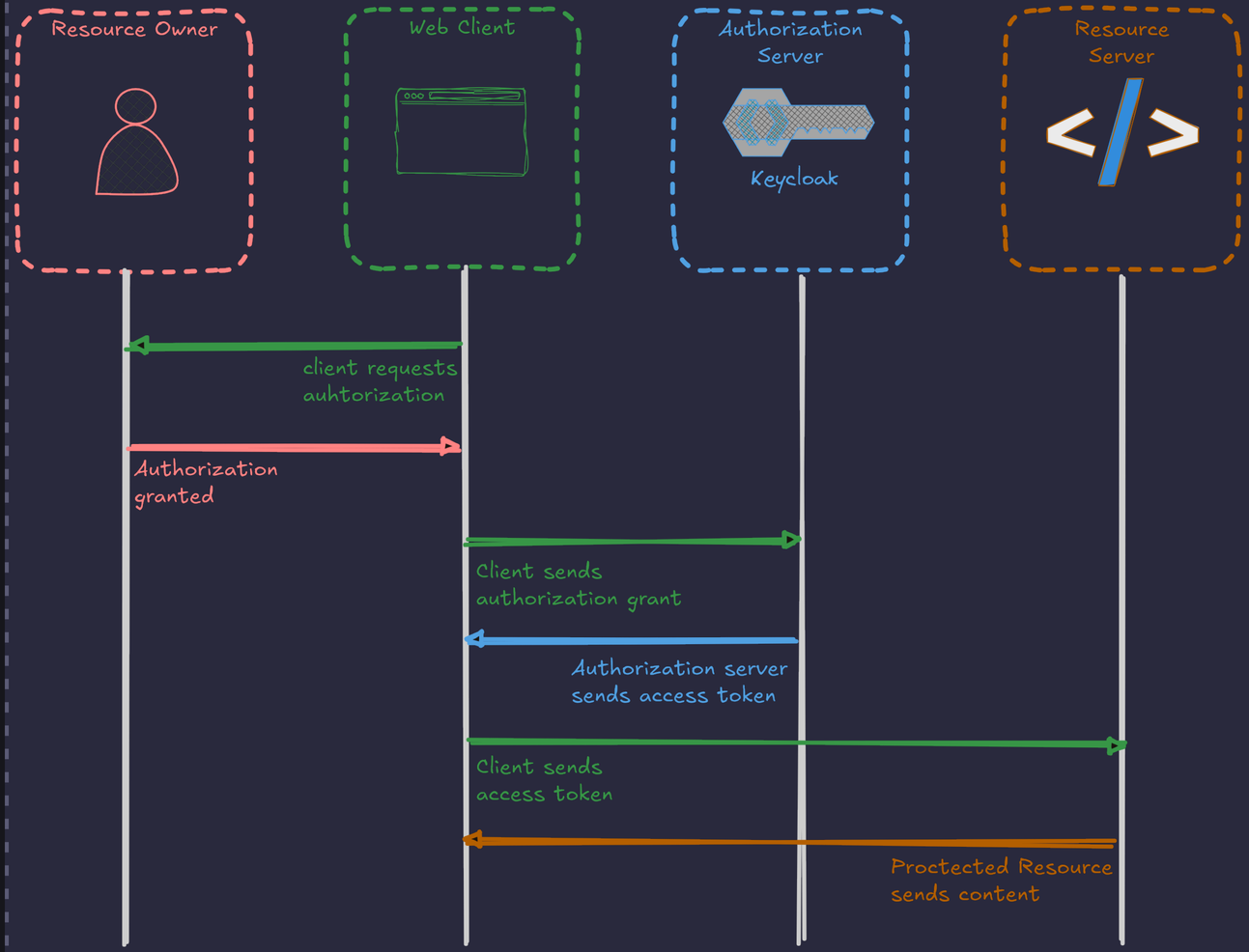
October 2012

Comparison with DevOpsBeerer

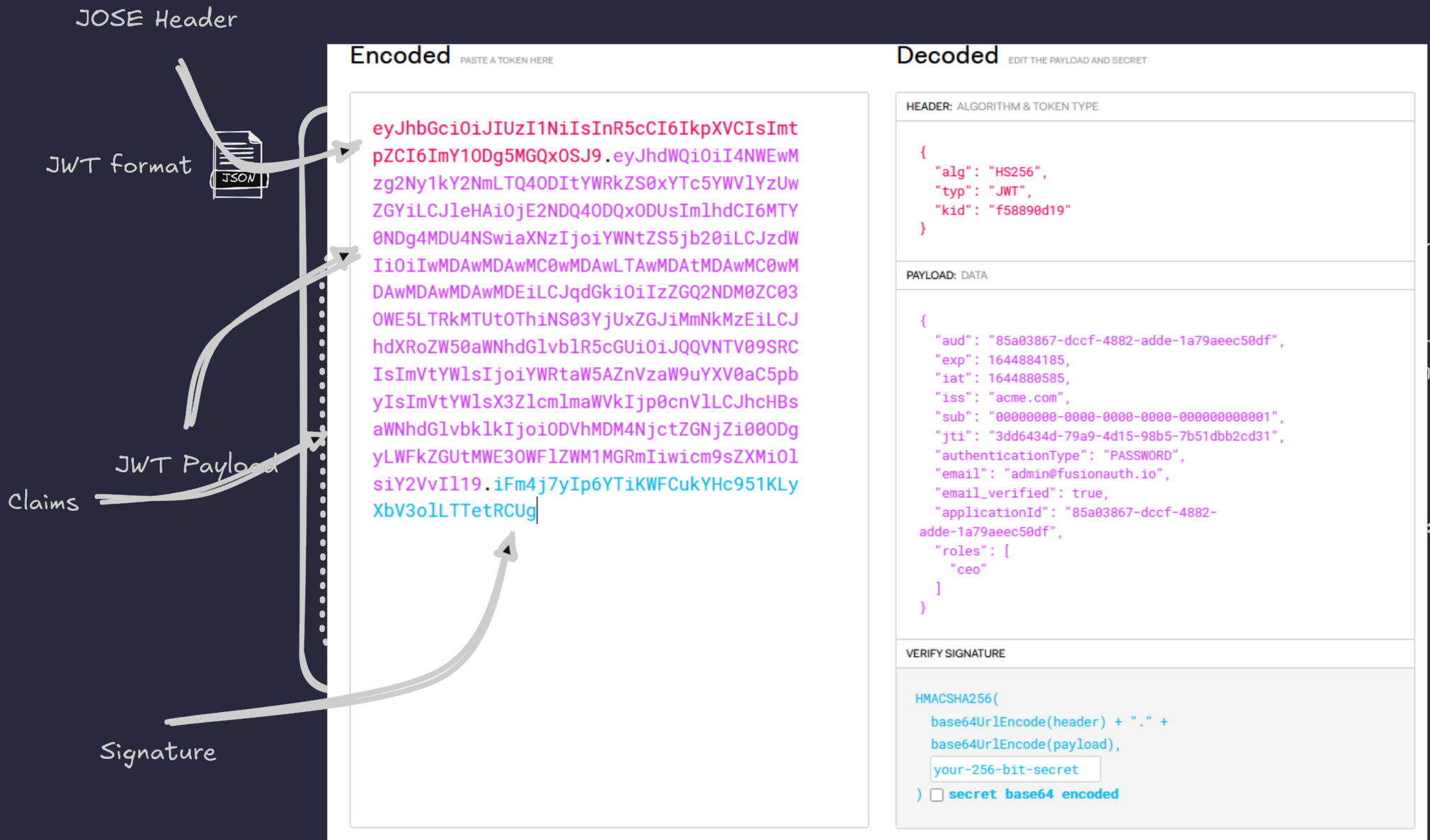


OAuth 2.0 : A Matter of Tokens

- The **Valet Key** principle :
  - Grant only needed permissions instead of full scope (scoped permissions).
  - One key per application.
- **Access Token :**
  - Delegated authorization , short time based.
- **Refresh Token :**
  - Can renew access token from AS.

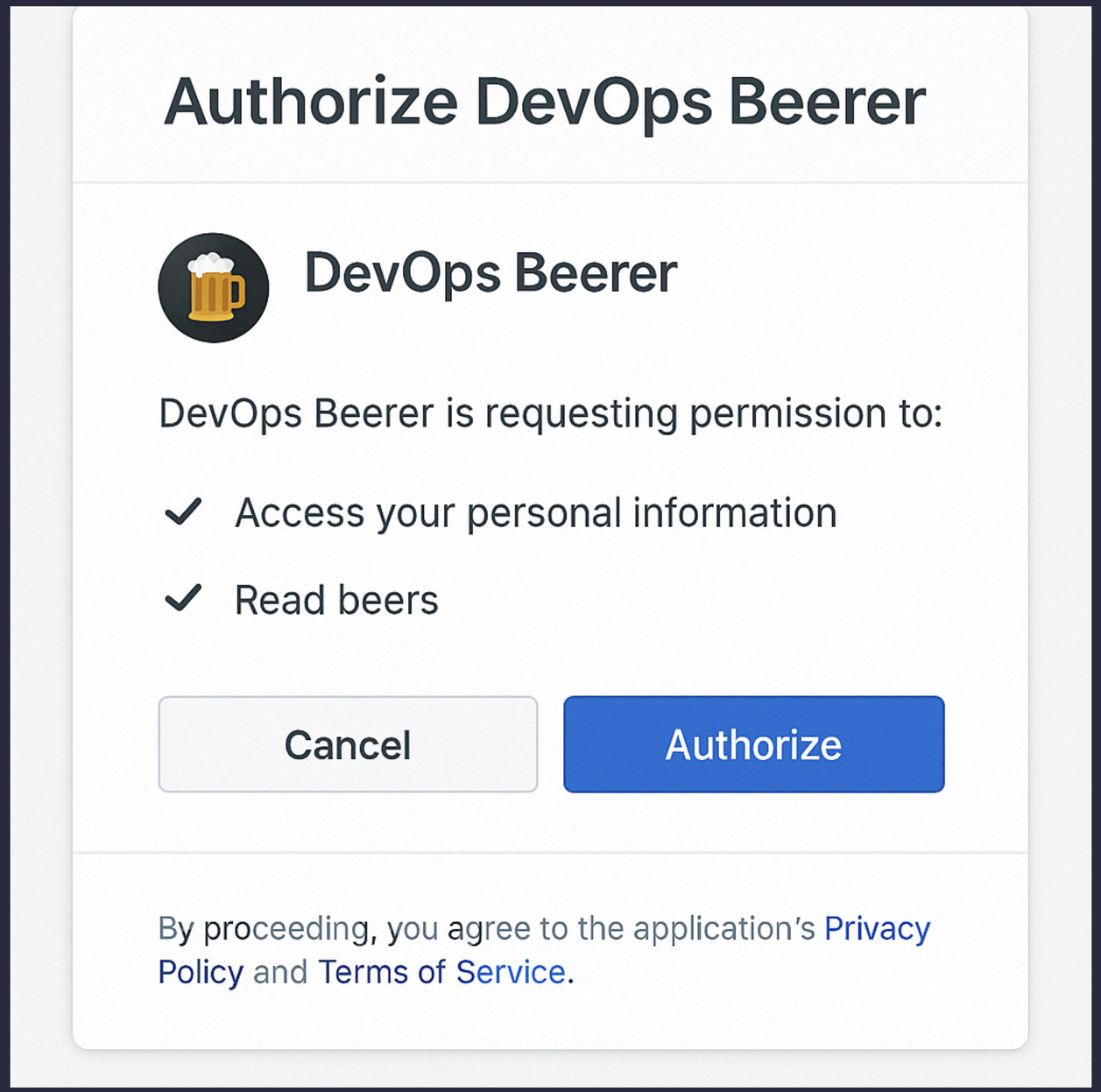


OAuth 2.0 : Anatomy of Access Token



OAuth 2.0 : Also Scopes...

- A sets of rights for a protected resource :
  - Limit API Access.
  - Displayed in user consent.
  - Scopes are additive in nature.
- OAuth does not define values, it's related to the API design.
- Some scopes will unlock certain claims (openid)



The Authorization Server

- Also know as : **IdP** (**Id**entity **P**rovider), for authentication purposes.
- **Authenticates resource owners**
  - Multiple methods available (x509, passwords, TOTP).
- **Manages the permissions/scopes**
  - Can renew access token from AS.

Auth0 ZITADEL Ping Identity

authentik KEYCLOAK

Authorization server endpoints

/authorize	→	Starts the auth process
/token	→	Forges tokens
/userinfo	→	(OIDC) User infos
/.well-known/*	→	Configuration discovery
/introspect	→	Checks token validity
/revoke	→	Revokes a token

OAuth 2.0 : Interactions

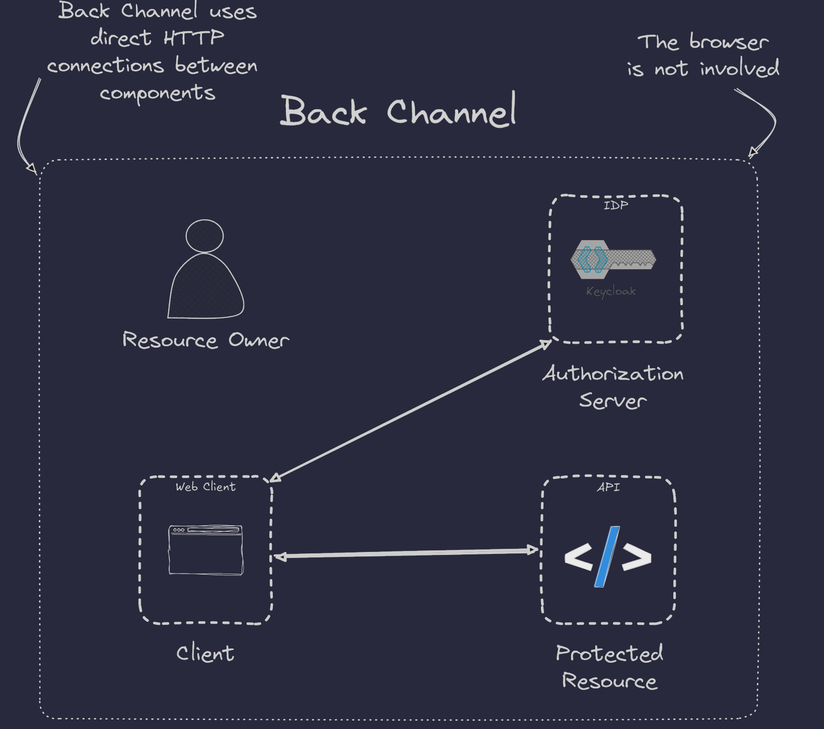
Back Channel vs. Front Channel

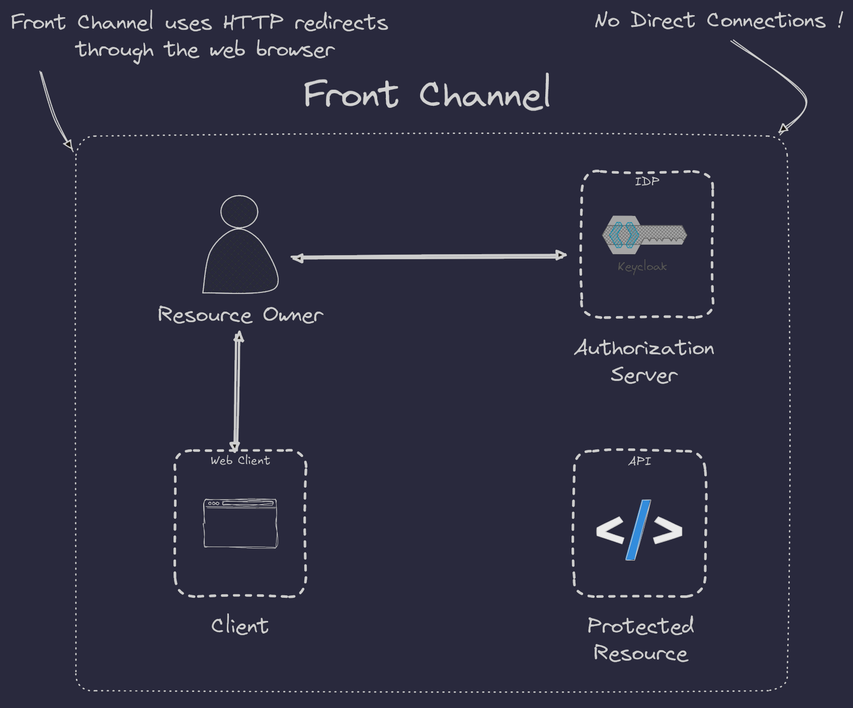
Back Channel:

- Communications outside of the user agent.
- Using HTTP Headers, bodies

Front Channel :

- Used when two components could not connect directly to each other.
- Using URL Parameters, non sensitive data.
- Using browser redirects.



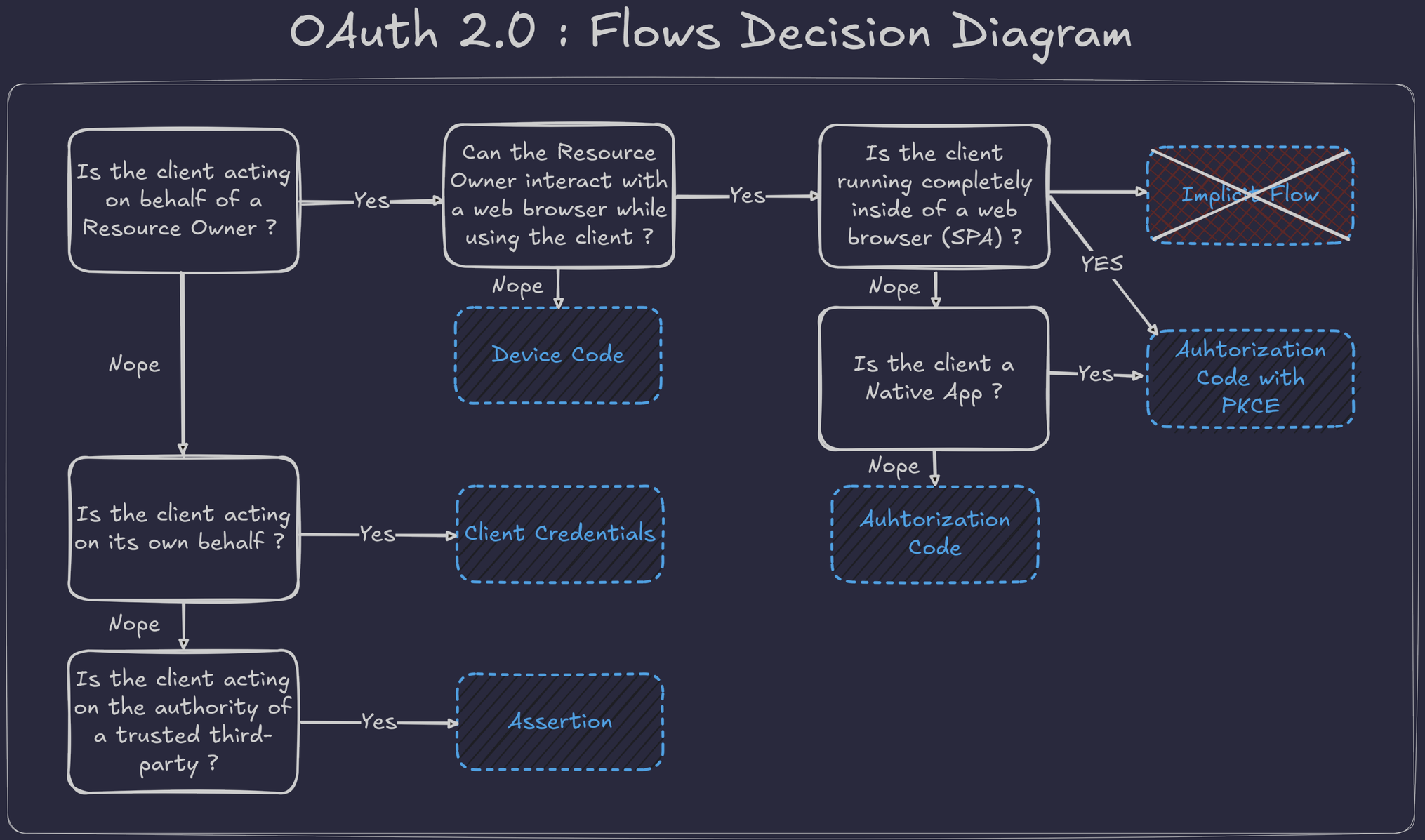


OAuth 2.0 : Client types

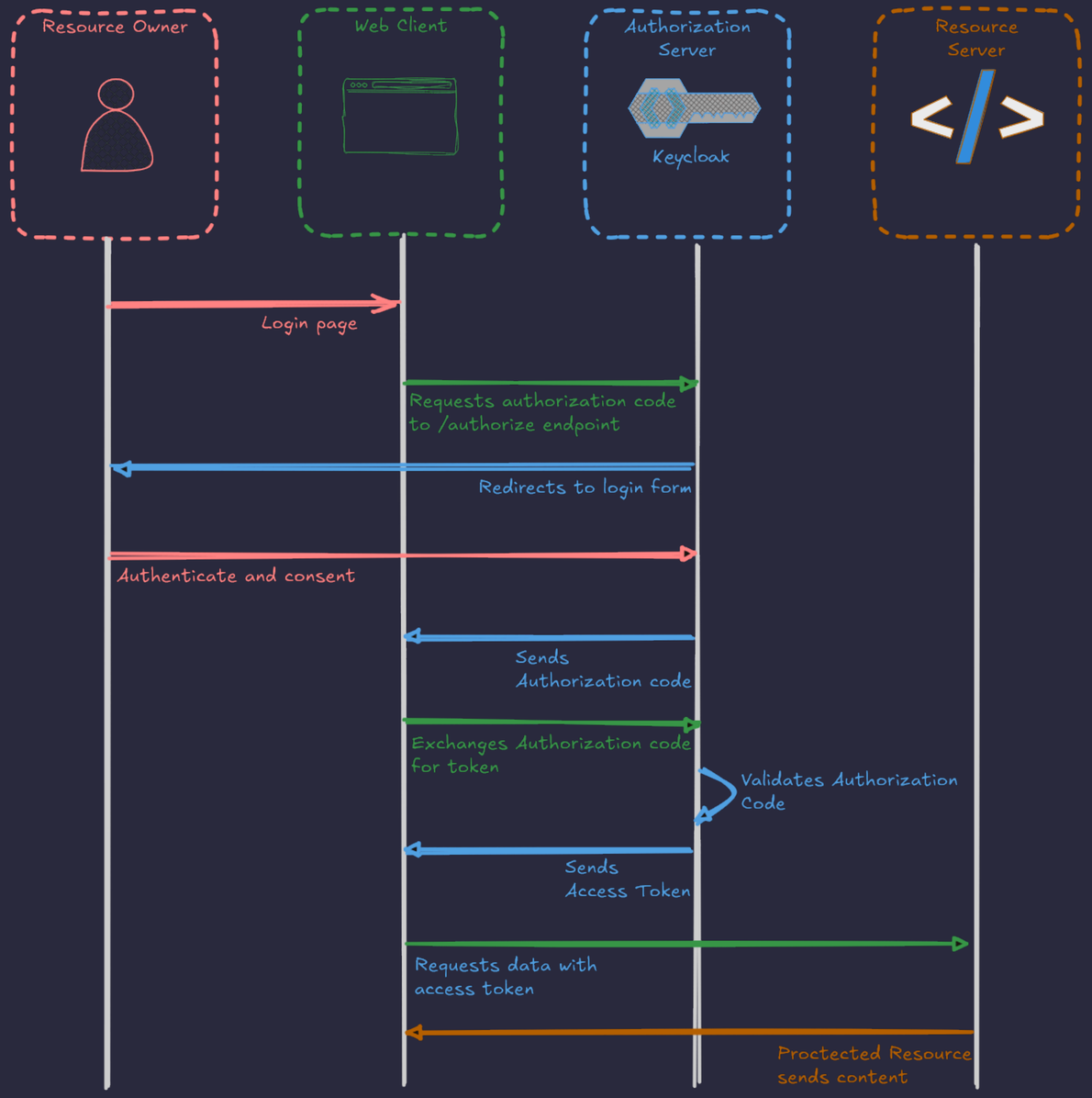
Confidential vs Public

- **Public** : Unable to register client secrets (CSR/JS, native apps)
- **Confidential** : Can authenticate securely with the Autorization Server (SSR apps).

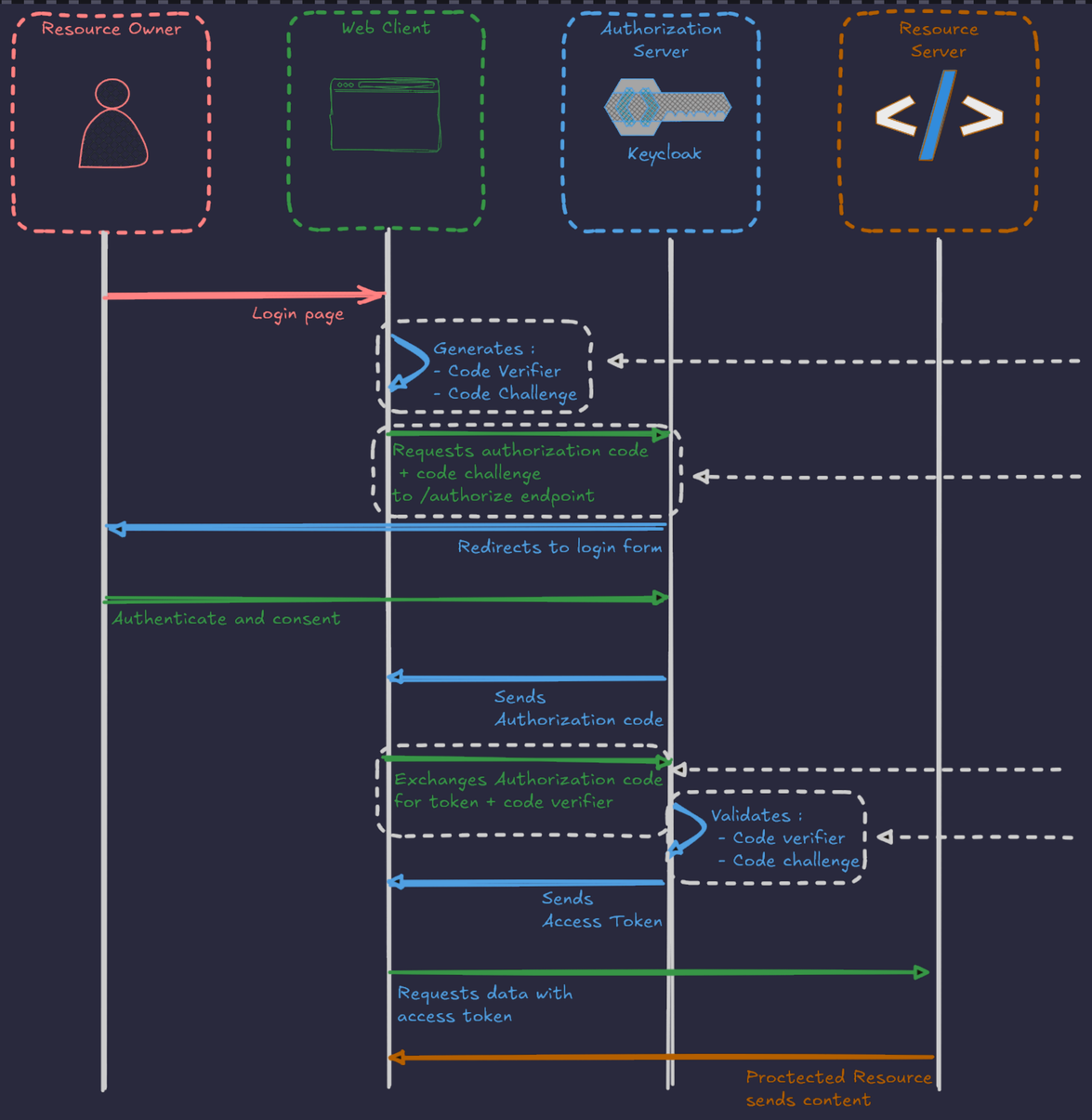
OAuth 2.0 : Flows / Grant Types



OAuth 2.0 : Authorization code grant



OAuth 2.0 : Authorization code grant with PKCE



OAuth 2.0 : Is Not

- **An authentication protocol**
  - Could be used to build one.
  - No user information provided

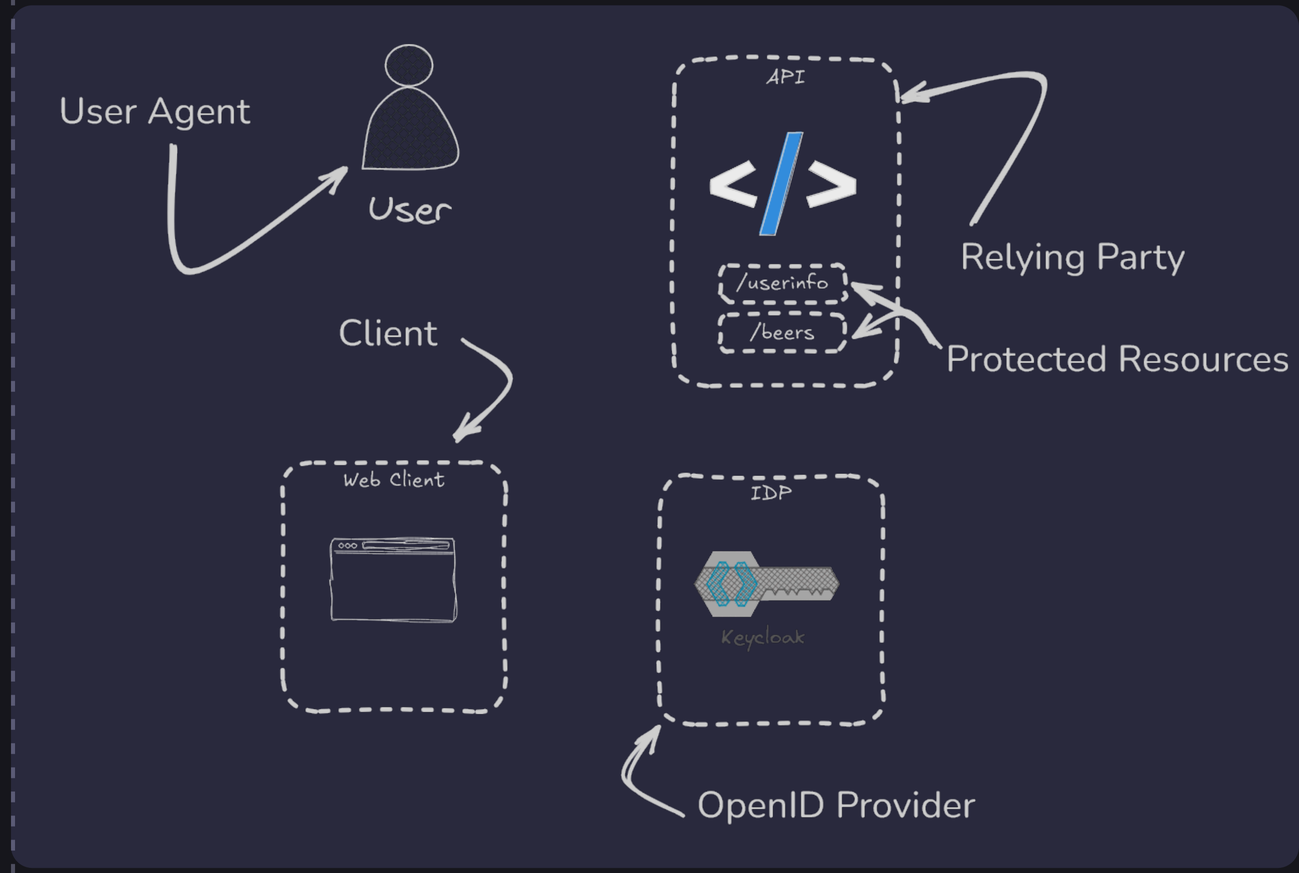
- Does not define **authorization mecanism**:
  - It provides “authorization delegation”.
  - It doesn’t define the content of AuthZ.
  - Up to the API (scopes)

- Not a **single protocol but a specification**.
  - Framework on grant types (aka. flows)
  - Leaves room for flexibility
  - Does not define a Token format

OpenID Foundation
ISO/IEC 26131:2024

OpenID Connect : An Authentication Protocol

- Federate Users using the same Identity Provider UserInfo API.
- Mostly used for Web application (internet)
- Delivers an **ID Token** : User Information



February 2014

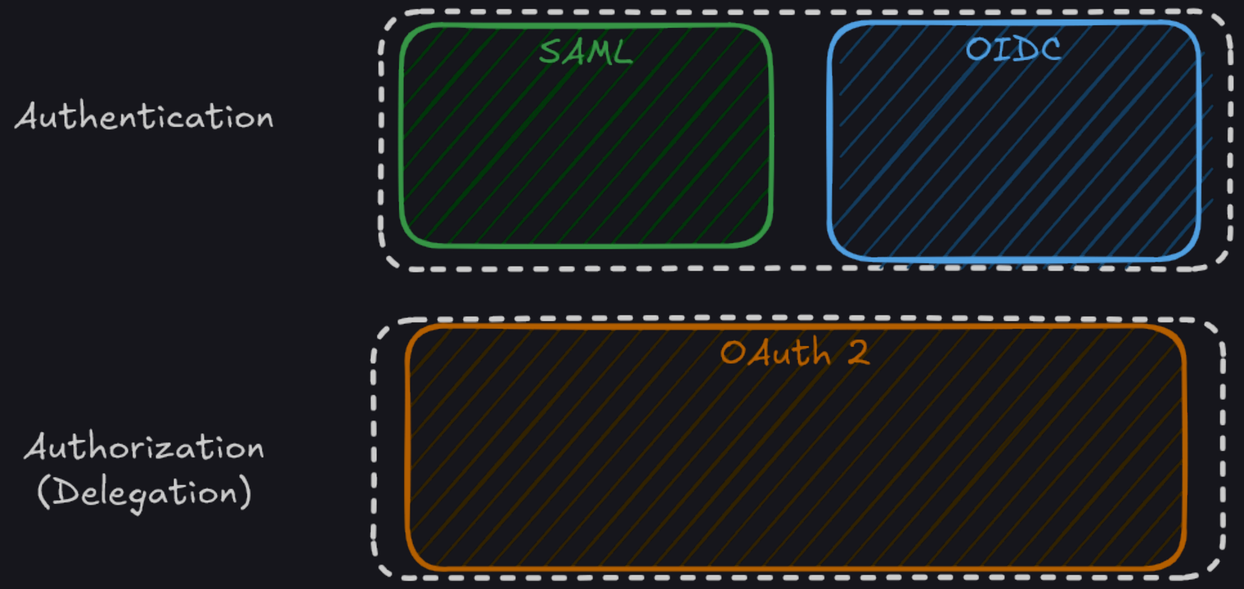
OASIS... is good

What about SAML ?

- SAML Authenticates users just as OIDC does
- SAML is mature, generally used for enterprise apps.
- Can be used on top of OAuth with Assertion Flow.
- Uses XML Assertions instead of JWT Tokens.

Let’s recap

- OAuth is for Access Delegation
- OIDC and SAML are used for Authentication



ENTERPRISES NEEDS OF APPLICATION SECURITY

MFA and policy enforcment

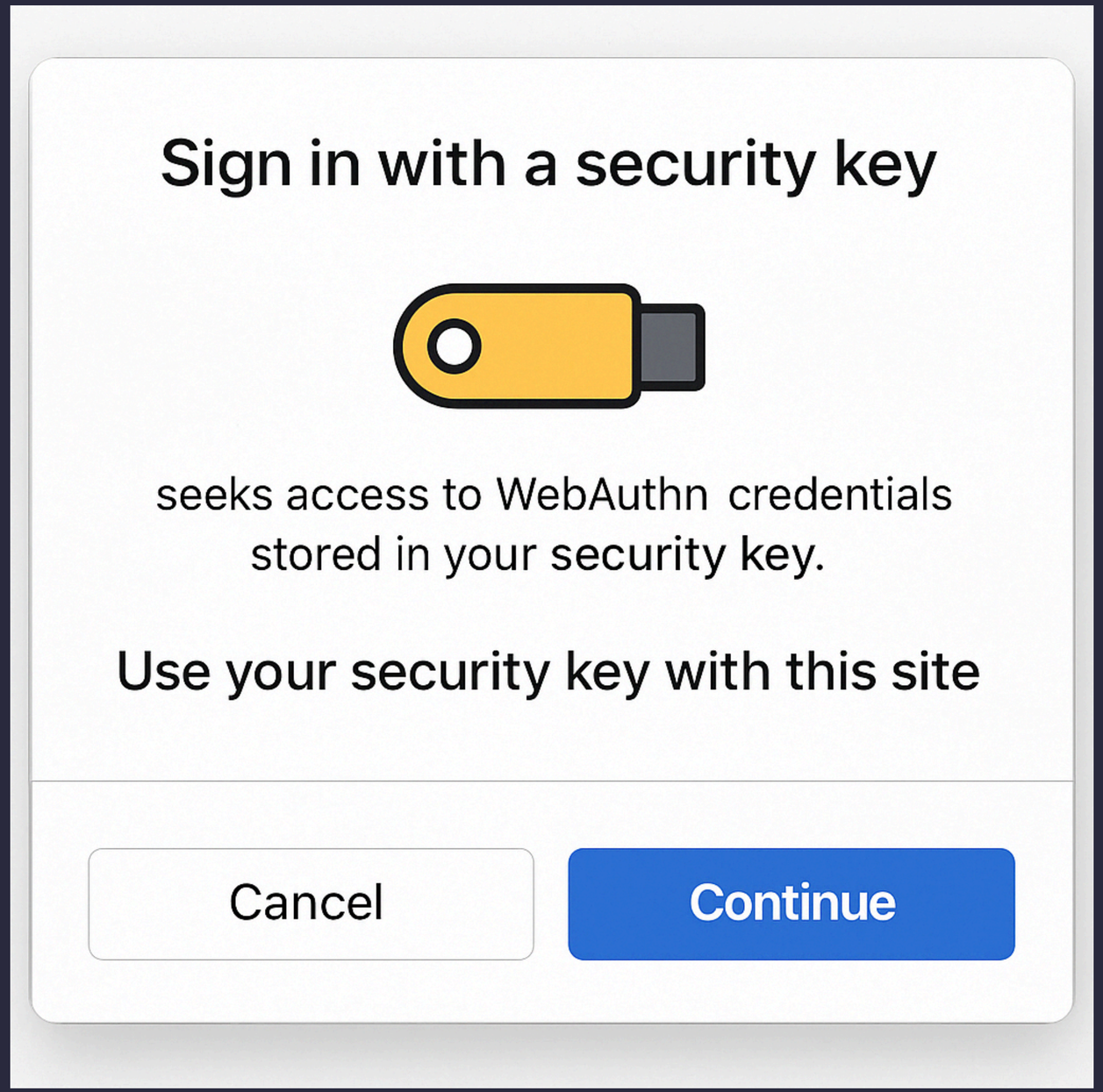
- Add policies to enforce security guards. Enable MFA to reduce password leaks usage.
- Even with my password leaked, no one can stole my beers without my mobile phone !

FIDO Alliance

Passwordless : WebAuthN

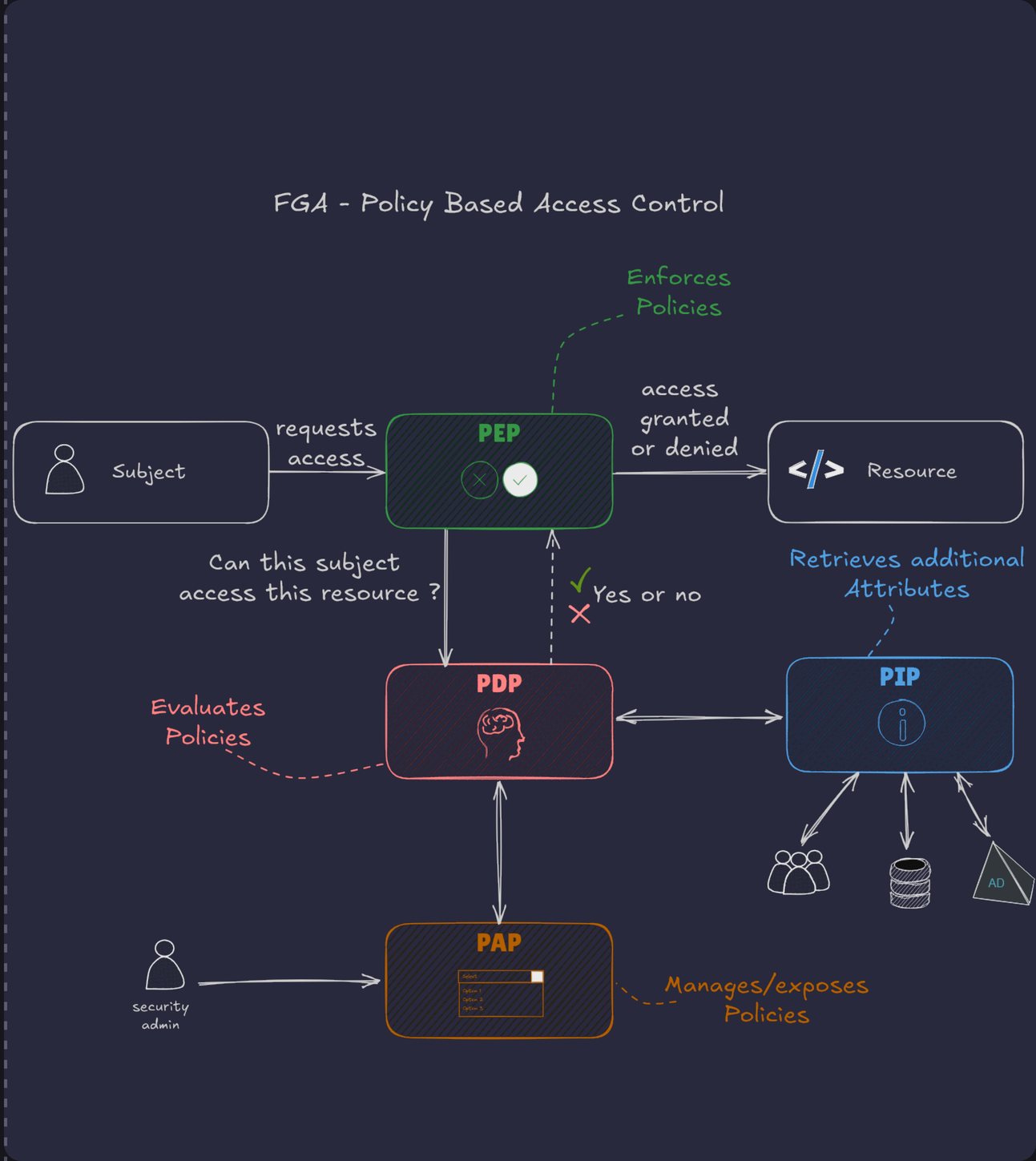
- Core component of the FIDO2 Spec, extension of the API Credential Management (asym crypto)
- Works with both security keys and TPM (Trusted Platform Module) inside laptops/mobiles.

https://webauthn.io/



What's PBAC ?

- Rely on policies engines, ex : OPA, AWS Cedar..
- Allows **decoupling** code from authz.
- ABAC can rely on PBAC :
  - **Conditions :** Attributes of subjects
  - **Relationship :** Connections between subjects
- The four horsemen of PBAC :
  - **PEP :** Policy Enforcement Point
  - **PDP :** Policy Decision Point
  - **PIP :** Policy Information Point
  - **PAP :** Policy Administration Point



The Future

FUTURE

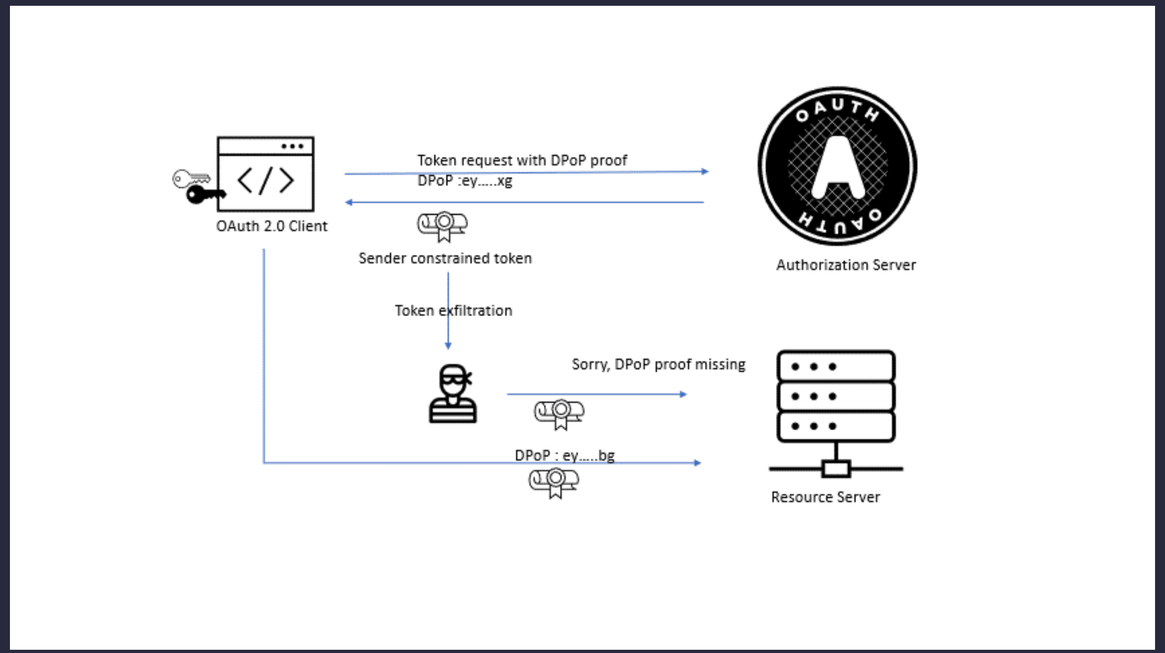
RFC #9449

Beyond Beerer Tokens ;)

Bearer can be intercepted, and reused ..

The need to demonstrated the Proof of Possession

```
GET /protectedresource HTTP/1.1
Host: resource.example.com
Authorization: DPoP <Access Token>
DPoP: <DPoP Token>
```



Q&A

Workshop

CONTEXT

Workshop Steps

- Step 1 : App without authentication.
- Step 2 : Implement OAuth2.0
- Step 3 : Protect routes with roles.
- Step 4 : Add some scopes

DEVOPSBEERER

CONTEXT

Keycloak IAM

- Authorization Server for OAuth2.
- Identity Provider for OIDC .

WORKSHOP

GitHub Repository

👉 Scan the QR Code to Access

👉 https://github.com/tintin92350/devopsdays-2025-workshop-apis-authz-authn