Identity Access Management

Sommaire

Rappel IAM
Active directory
IAM vs AD
Fonctionnement IAM
SAML
OAUTH 2
OIDC

•**Qu’est-ce que l’IAM ?**

•*IAM (Identity & Access Management)* = Gestion des identités numériques et des accès aux ressources.

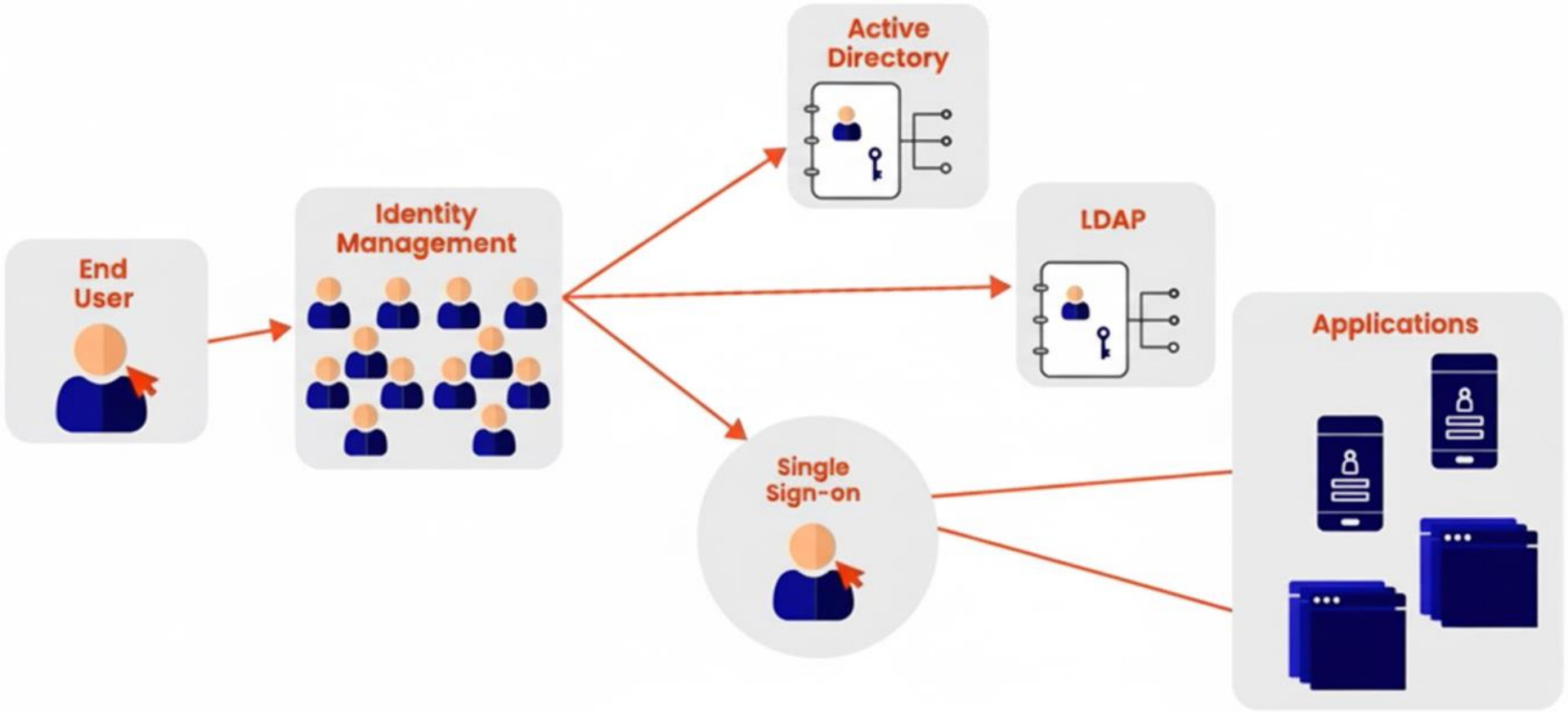
•Objectifs principaux :

- **Sécurité** → empêcher les accès non autorisés
- **Conformité** → respecter les règles et réglementations
- **Productivité** → simplifier l’expérience utilisateur (SSO)

•**Acteurs clés :**

- **Utilisateurs** (collaborateurs, partenaires, clients)
- **Applications** (SaaS, on-premise, mobiles)
- **Ressources** (fichiers, bases de données, systèmes)

SSO implementation



Étape 1 : l’identité est gérée
L’utilisateur passe d’abord par le système de gestion des identités (role, groupe, utilisateur)

Étape 2 : le SSO entre en jeu
Une fois l’identité vérifiée, le SSO permet à l’utilisateur de se connecter une seule fois et d’accéder aux ressources

Le SSO agit comme un pont entre l’identité de l’utilisateur et les différentes applications.
Il transmet un jeton sécurisé qui prouve que l’utilisateur est bien authentifié.

Étape 3 : accès aux applications (mobiles,web, locale)

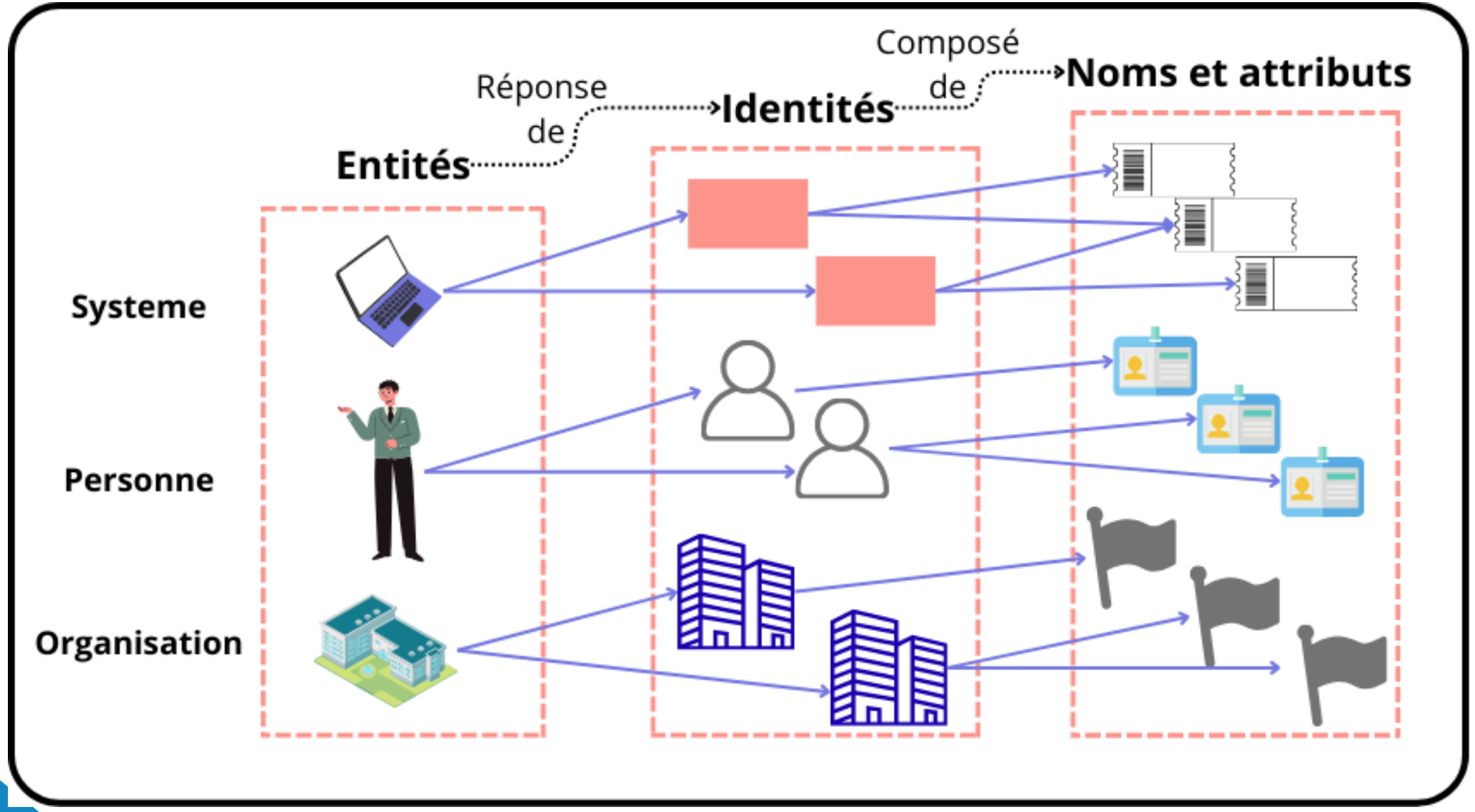
Active Directory

- **Active Directory (AD)** = annuaire d’entreprise développé par Microsoft
- **Fonctionne en local** (on-premise), intégré aux environnements Windows
- Gère les **utilisateurs, groupes, mots de passe, droits d’accès**
- Très utilisé dans les infrastructures traditionnelles
- **Limité pour les environnements cloud, mobiles ou hybrides**
- Nécessite des extensions pour gérer le SSO ou les identités externes

- L’IAM est venu **moderniser la gestion des identités** : plus flexible, plus sécurisé, plus adapté au cloud

Comment fonctionne l'IAM

Gestion des identités



•**Système** : machines, logiciels, comptes techniques
• **Personne** : individus, collaborateurs, utilisateurs finaux
•**Organisation** : entreprises, départements, partenaires

•**Relation entre les éléments** :
•Une **entité** possède une ou plusieurs **identités**
•Chaque identité est **composée de noms et d'attributs** : prénom, rôle email, niveau d'accès

• **Noms et attributs** :
•Ce sont les **données descriptives** qui permettent d'identifier et de qualifier une identité

Flux logique :
•L'identité est une **réponse à une entité**
•Elle est **composée d'attributs** qui permettent de prendre des décisions d'accès

Les Roles

- **1 RBAC:**
  est un modèle de contrôle d'accès basé sur les **rôles**.
  Chaque utilisateur se voit attribuer un ou plusieurs rôles, et chaque rôle correspond à un ensemble de permissions.

- **Exemple** :
  Rôle Comptable ⟶ accès aux factures.
  Rôle Administrateur RH ⟶ accès complet aux dossiers du personnel.
  Un utilisateur qui a le rôle Comptable hérite automatiquement des permissions liées.

Les Roles

- **2 ABAC:**

est un modèle de contrôle d'accès basé sur des **attributs** (de l'utilisateur, de la ressource, de l'action ou du contexte).

Les règles sont définies sous forme de politiques logiques.

- **Exemple** :

Condition : si department = Finance **et** country = FR → accès autorisé au reporting financier.

Condition : si clearance = Secret **et** heure < 18h → accès aux documents sensibles.

RBAC

Role Based Permissions

Example: Admin Role can create and delete files

Users can update and read files

ABAC

Attribute Based Permissions

Attributes can be Location, Time, & Department

Example: Worker in Finance has access from 9-5pm

Protocol SSO

Principe et composants

Identité unique : l’utilisateur est authentifié une fois via un fournisseur d’identité (IdP)

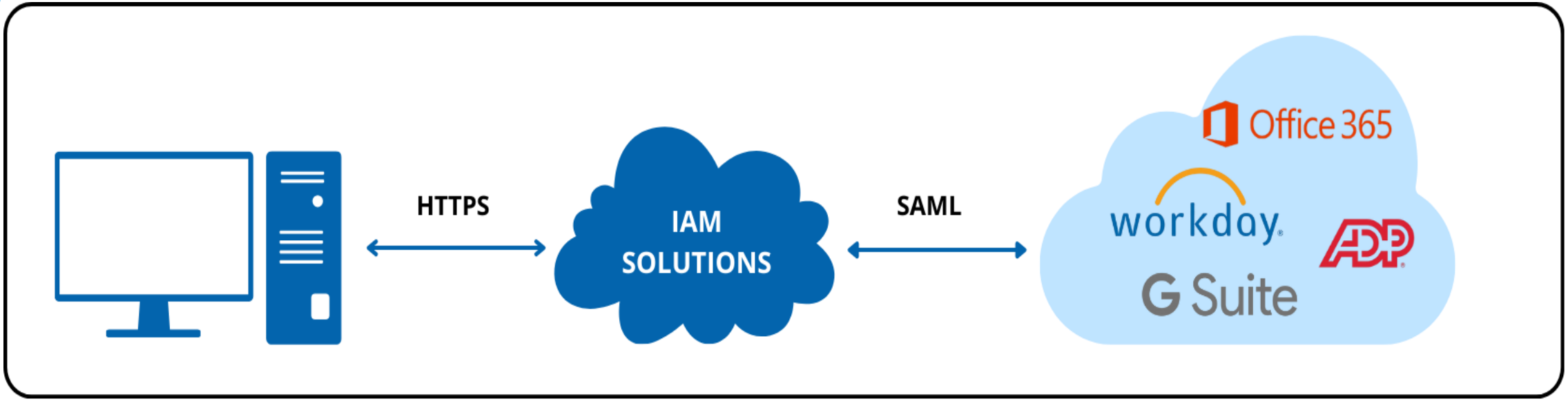
Jeton d’authentification : une preuve numérique (token, assertion) transmise aux applications

Fournisseur d’identité (IdP) : entité qui vérifie l’identité (ex : Azure AD, Okta, Google)

Fournisseur de service (SP) : application ou ressource à laquelle l’utilisateur veut accéder

Protocole d’échange : mécanisme qui permet de transmettre l’identité (SAML, OAuth2, OIDC)

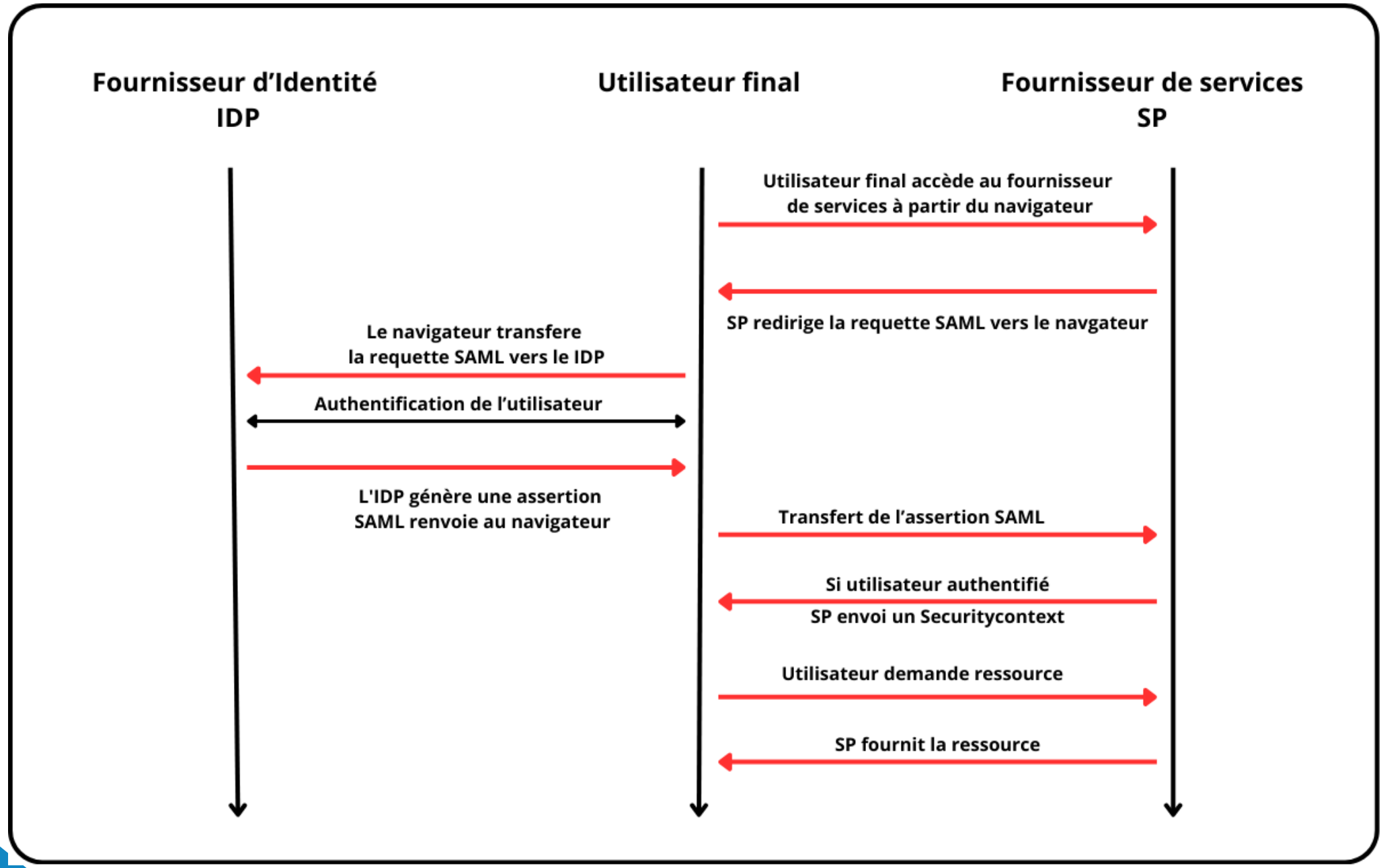
SAML



SAML

SAML (Security Assertion Markup Language) est un **protocole d'authentification** basé sur le langage XML, qui permet à un utilisateur de se connecter à plusieurs applications via un mécanisme de **Single Sign-On (SSO)**.

Il fonctionne en transmettant une **assertion** (une preuve d'identité) entre un **fournisseur d'identité (IdP)** et un **fournisseur de service (SP)**, sans que l'utilisateur ait à ressaisir ses identifiants.


Fournisseur d'Identité
IDP
Utilisateur final
Fournisseur de services
SP
Utilisateur final accède au fournisseur
de services à partir du navigateur
SP redirige la requette SAML vers le navgateur
Le navigateur transfere
la requette SAML vers le IDP
Authentification de l'utilisateur
L'IDP génère une assertion
SAML renvoie au navigateur
Transfert de l'assertion SAML
Si utilisateur authentifié
SP envoi un Securitycontext
Utilisateur demande ressource
SP fournit la ressource

Oauth2

OAuth2 (Open Authorization)

Protocole d'**autorisation**

Permet à une application d'accéder à des ressources au nom d'un utilisateur

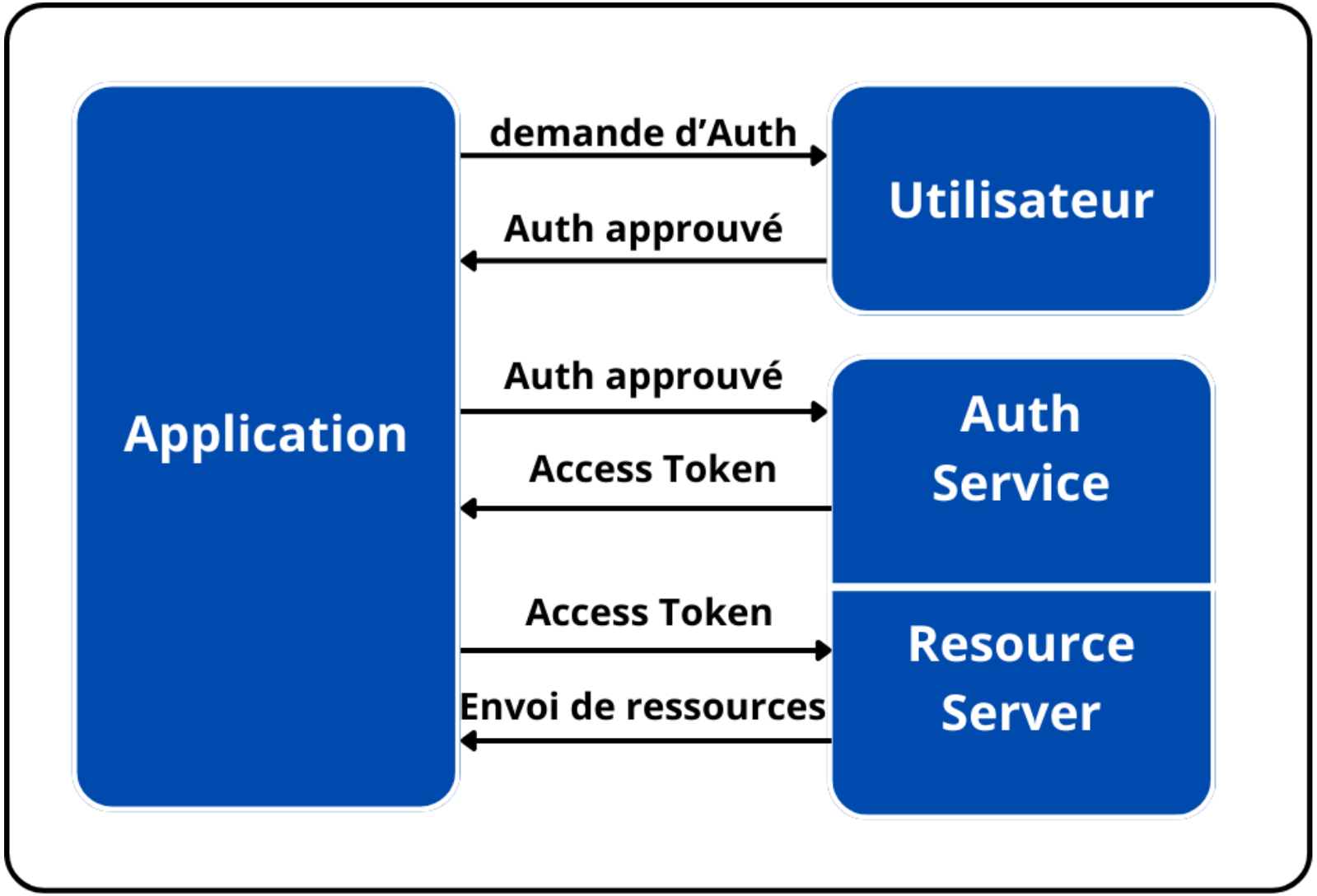
Ne fournit pas d'informations sur l'identité de l'utilisateur

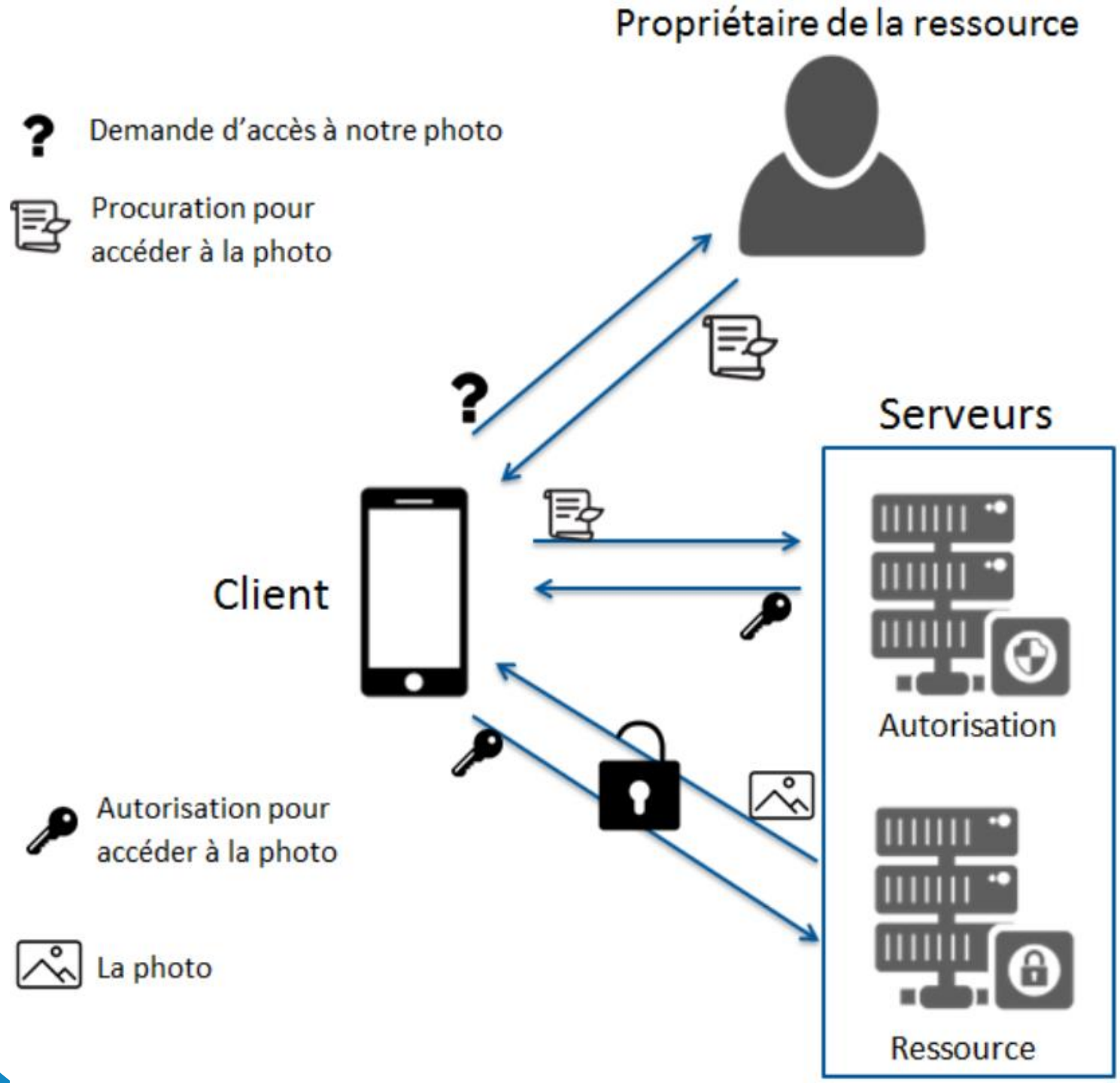
Utilisé pour les accès API, les applications mobiles, les services tiers

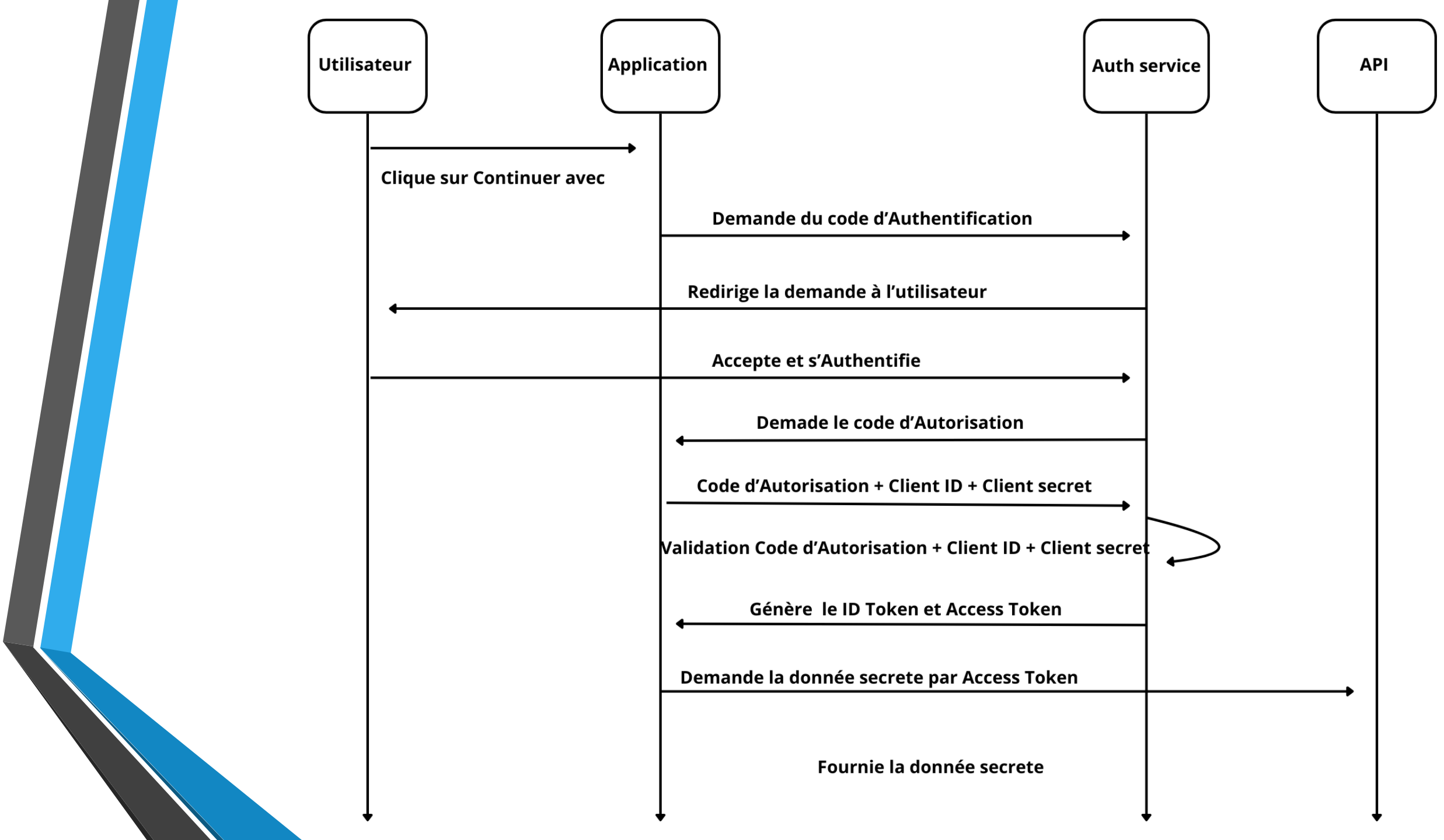
OpenID Connect

OIDC (OpenID Connect)
Extension d'OAuth2 qui ajoute l'**authentification**
Fournit un **ID Token** (JWT) contenant des infos sur l'utilisateur : nom, email, identifiant
Permet à une application de **savoir qui est l'utilisateur** et de le connecter
Utilisé pour le SSO, les connexions sécurisées aux applications web

OAUH2 / OIDC


Application
demande d'Auth
Auth approuvé
Utilisateur
Auth approuvé
Access Token
Auth Service
Access Token
Envoi de ressources
Resource Server


Propriétaire de la ressource
Demande d'accès à notre photo
Procuration pour accéder à la photo
Serveurs
Client
Autorisation
Autorisation pour accéder à la photo
La photo
Ressource


Utilisateur
Application
Auth service
API
Clique sur Continuer avec
Demande du code d'Authentification
Redirige la demande à l'utilisateur
Accepte et s'Authentifie
Demade le code d'Autorisation
Code d'Autorisation + Client ID + Client secret
Validation Code d'Autorisation + Client ID + Client secret
Génère le ID Token et Access Token
Demande la donnée secrete par Access Token
Fournie la donnée secrete