AYI NEDJIMI Consultants

SAML vs OIDC vs OAuth2 : choisir le bon protocole SSO

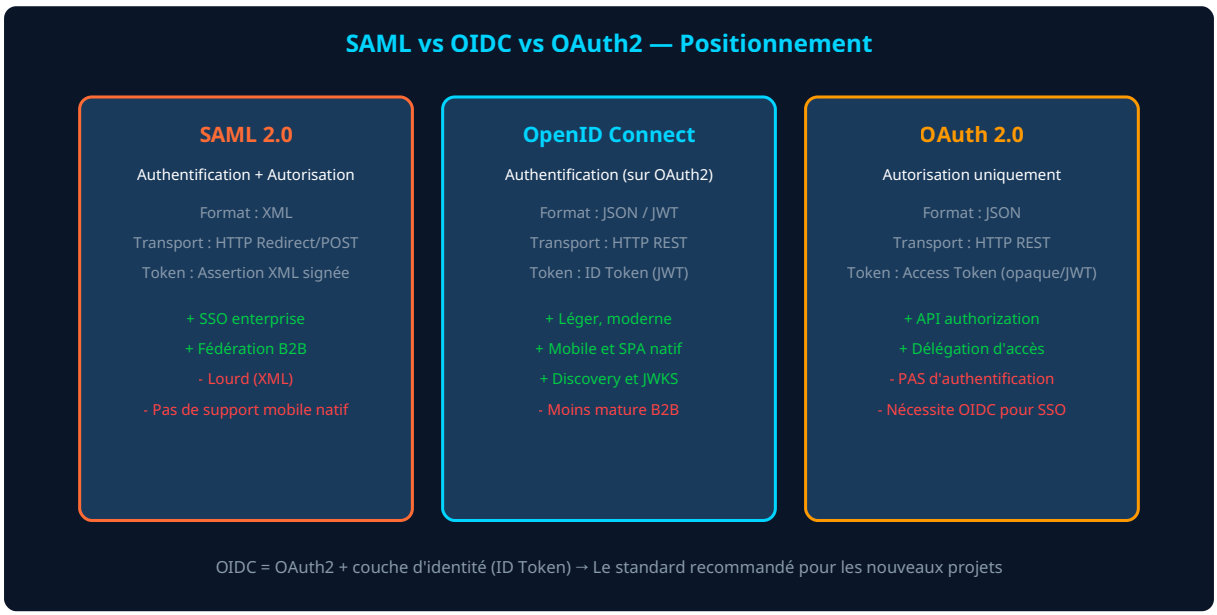
Catégorie : IAM et Gestion des Identités Lecture : 6 min Publié le : 12/03/2026 Auteur : Ayi NEDJIMI

SAML, OpenID Connect et OAuth2 : comparaison technique détaillée pour choisir le bon protocole de fédération d'identités selon vos cas d'usage en.

SAML, OpenID Connect, OAuth2 — trois protocoles qui reviennent systématiquement dans tout projet de fédération d'identités, et qui génèrent une confusion persistante. OAuth2 n'est pas un protocole d'authentification. OIDC est construit sur OAuth2 mais fait bien plus. SAML existe depuis 2005 et reste omniprésent dans les SI d'entreprise. Pour l'architecte sécurité ou l'ingénieur IAM, choisir le bon protocole pour chaque cas d'usage est une compétence fondamentale. Ce guide vous fournit une comparaison technique approfondie de ces trois standards, avec des arbres de décision concrets pour chaque scénario courant : SSO web, API-to-API, applications mobiles, fédération B2B et intégration legacy. Nous détaillerons les flux d'authentification, les différences de sécurité, les implications opérationnelles et les pièges d'implémentation que nous avons rencontrés sur le terrain. L'objectif est de vous rendre autonome dans le choix du protocole adapté à chaque situation, sans dépendre d'un vendeur qui vous orientera systématiquement vers sa solution.

Points clés à retenir

- **OAuth2** est un framework d'autorisation, pas d'authentification — ne l'utilisez jamais seul pour du SSO
- **OpenID Connect** (OIDC) ajoute la couche d'authentification à OAuth2 — c'est le standard moderne pour le SSO
- **SAML 2.0** reste pertinent pour les intégrations enterprise legacy et la fédération B2B
- Pour les nouvelles applications, OIDC doit être le choix par défaut
- La coexistence SAML + OIDC est la norme dans les SI d'entreprise hybrides



OAuth2 : un framework d'autorisation, pas d'authentification

La confusion la plus répandue : utiliser **OAuth2** seul pour authentifier des utilisateurs. OAuth2 est un framework d'autorisation délégué. Son objectif initial : permettre à une application tierce d'accéder aux ressources d'un utilisateur sans connaître son mot de passe. Quand vous autorisez une application à accéder à vos photos Google, c'est OAuth2. L'application reçoit un *access token* qui l'autorise à lire vos photos, mais ce token ne contient aucune information fiable sur votre identité.

Le problème survient quand des développeurs utilisent OAuth2 seul pour du SSO. Ils récupèrent un access token, appellent un endpoint /userinfo et considèrent l'utilisateur authentifié. Cette approche présente des failles de sécurité documentées : confused deputy attack, token injection, absence de validation du destinataire du token. La documentation OAuth.net explique en détail pourquoi OAuth2 seul n'est pas un protocole d'authentification. Pour l'authentification, utilisez OIDC qui est construit sur OAuth2 avec les garanties de sécurité nécessaires.

OpenID Connect : le standard moderne pour l'authentification

OpenID Connect (OIDC) ajoute une couche d'identité au-dessus d'OAuth2. La différence clé : en plus de l'access token, le serveur d'autorisation délivre un *ID Token* au format JWT signé qui contient les claims d'identité de l'utilisateur (sub, name, email, aud, iss, exp). Ce token est signable et vérifiable par le client sans appel réseau supplémentaire, grâce aux clés publiques publiées via le **JWKS endpoint**.

OIDC introduit aussi le mécanisme de **Discovery** : chaque provider publie un document JSON à l'URL `/.well-known/openid-configuration` qui décrit tous ses endpoints (authorization, token, userinfo, jwks_uri). Cette auto-configuration simplifie radicalement l'intégration. Les flows OIDC recommandés en 2026 : **Authorization Code + PKCE** pour les applications web et mobiles, **Client Credentials** pour les communications machine-to-machine. Le flow Implicit est déprécié par les recommandations IETF.

SAML 2.0 : le vétéran qui refuse de prendre sa retraite

SAML 2.0 (Security Assertion Markup Language) est le standard historique de fédération d'identités en entreprise. Publié en 2005, il reste omniprésent dans les SI des grandes organisations. Les applications enterprise (SAP, Salesforce, ServiceNow, Workday) supportent toutes SAML. Les partenariats B2B entre organisations s'appuient massivement sur la fédération SAML. Le format XML des assertions est verbeux mais extensible, et la signature XML garantit l'intégrité et l'authenticité.

Le flux SAML SP-Initiated (le plus courant) fonctionne ainsi : l'utilisateur accède à l'application (Service Provider). Le SP génère une *AuthnRequest* XML et redirige vers l'IdP (Identity Provider). L'IdP authentifie l'utilisateur, génère une **Assertion SAML** signée contenant les attributs de l'utilisateur et la renvoie au SP via le navigateur (HTTP POST). Le SP valide la signature et crée une session locale. Les configurations avancées Entra ID supportent SAML comme protocole de SSO pour les applications enterprise.

Comparaison technique détaillée

Critère	SAML 2.0	OIDC	OAuth 2.0
Fonction principale	Authentification + SSO	Authentification + SSO	Autorisation déléguée
Format des tokens	XML (Assertion)	JWT (ID Token)	Opaque ou JWT
Transport	HTTP Redirect, POST	HTTP REST	HTTP REST
Taille du token	2-10 KB	0.5-2 KB	Variable
Support mobile	Limité	Natif	Natif
API protection	Non adapté	Oui (via access token)	Oui
Discovery	Metadata XML manuelle	.well-known auto	Non standard
Maturité B2B	Excellente	En progression	Limitée
Complexité d'implémentation	Élevée	Moyenne	Faible

Arbre de décision : quel protocole pour quel cas d'usage

Le choix du protocole dépend du scénario technique. Pour une **nouvelle application web interne** : OIDC avec Authorization Code + PKCE. Pour une **application mobile** : OIDC avec Authorization Code + PKCE (jamais Implicit). Pour la **protection d'API REST** : OAuth2 avec des access tokens JWT validés par la ressource. Pour l'**intégration d'une application SaaS enterprise** (Salesforce, SAP) : SAML si c'est le seul protocole supporté, OIDC si disponible. Pour la **fédération B2B** avec un partenaire : SAML reste le standard le plus largement supporté.

Pour les environnements hybrides où coexistent des applications SAML et OIDC, votre architecture Zero Trust IAM doit supporter les deux protocoles via un Identity Provider multi-protocole. Entra ID, Okta et Ping Identity gèrent nativement la double fédération SAML/OIDC. Le connecteur Entra Connect ajoute la dimension hybride on-premise/cloud à cette architecture.

Pièges d'implémentation et vulnérabilités courantes

Chaque protocole a ses vulnérabilités spécifiques. En **SAML** : les attaques par XML Signature Wrapping exploitent une validation incorrecte de la signature XML. La recommandation : utilisez des bibliothèques maintenues (OneLogin SAML toolkit, Spring Security SAML) et validez systématiquement le schema XML. Les injections d'attributs dans les assertions SAML sont un autre vecteur à surveiller.

En **OIDC/OAuth2** : le redirect URI doit être validé de manière stricte (pas de wildcards, pas de regex permissive). Le state parameter est obligatoire pour prévenir les attaques CSRF. Le nonce parameter dans l'ID Token prévient les replay attacks. Le PKCE (Proof Key for Code Exchange) est obligatoire pour tous les clients publics. L'access token ne doit jamais être stocké dans le localStorage du navigateur (vulnérable au XSS) — utilisez des cookies HttpOnly avec le flag Secure et SameSite=Strict.

Migration progressive de SAML vers OIDC

La migration de SAML vers OIDC se fait application par application, sans big bang. Identifiez d'abord les applications qui supportent les deux protocoles (la plupart des SaaS modernes). Migrez-les en premier vers OIDC pour bénéficier de la simplification opérationnelle (auto-discovery, tokens légers, meilleur support mobile). Pour les applications legacy uniquement SAML, maintenez la fédération SAML — la coexistence des deux protocoles sur le même IdP est transparente.

Les gains de la migration : réduction de la taille des tokens (de 5 KB XML à 500 bytes JWT), support natif des applications mobiles et SPA, auto-configuration via Discovery qui simplifie le onboarding de nouvelles applications de 2 jours à 30 minutes. La journalisation centralisée des authentifications OIDC est aussi plus simple à parser et à corréler que les logs SAML XML.

Questions fréquentes sur les protocoles de fédération

Peut-on utiliser SAML et OIDC simultanément sur le même IdP ?

Oui, tous les IdP modernes (Entra ID, Okta, Ping Identity, Keycloak) supportent la fédération multi-protocole. Un même utilisateur peut accéder à une application SAML et une application OIDC via le même IdP avec un SSO transparent. L'IdP gère la traduction entre les protocoles en interne. Cette coexistence est la norme dans les SI d'entreprise et n'introduit pas de risque de sécurité additionnel.

OIDC va-t-il remplacer SAML à terme ?

À long terme, probablement. OIDC gagne du terrain chaque année, surtout dans les applications cloud-native et mobiles. Cependant, SAML reste solidement ancré dans l'écosystème enterprise avec des milliers d'intégrations existantes. La disparition de SAML n'est pas prévisible avant 2030 au minimum. La stratégie pragmatique : OIDC par défaut pour les nouveaux projets, SAML maintenu pour le legacy, migration opportuniste quand le support OIDC est disponible.

Comment sécuriser les tokens JWT en production ?

Cinq règles fondamentales : utilisez RS256 ou ES256 pour la signature (jamais HS256 en multi-tenant). Validez systématiquement les claims iss, aud, exp et nbf côté client. Gardez des durées de vie courtes (15 minutes pour les access tokens, 1 heure pour les ID tokens). Ne stockez jamais de données sensibles dans le payload JWT (les claims sont encodés, pas chiffrés). Utilisez le refresh token rotation pour limiter l'impact d'un token volé.

Sources et références : ANSSI · MITRE ATT&CK

Synthèse et recommandations

Le choix entre SAML, OIDC et OAuth2 n'est pas un dilemme — c'est une question de contexte. OIDC est le standard par défaut pour les nouvelles applications. OAuth2 protège vos API. SAML reste pertinent pour les intégrations enterprise et la fédération B2B. La maturité d'une organisation IAM se mesure à sa capacité à faire coexister ces protocoles de manière cohérente et sécurisée, avec un Identity Provider central comme point de contrôle unique. Maîtrisez les trois, déployez le bon au bon endroit.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr