WHITE PAPER | IDENTITY AND ACCESS ARCHITECTURE

Federation, Access Control Models, and Identity Standards

What federation, RBAC, ABAC, SAML, OAuth 2.0, OpenID Connect, SCIM, and LDAP do, the problems they solve, and how they work together across industries

Mark Adams
Identity security, governance, and security architecture
September 2026

Executive summary

Every organization has to answer three identity questions for every person, device, service, and AI agent: who exists, who you are, and what you are allowed to do. The standards in this paper are the industry's shared answers. Used together, they let one strong sign-in reach hundreds of applications, keep accounts in step with employment status, and let APIs make precise, auditable access decisions.

This paper explains federation, role-based and attribute-based access control (RBAC and ABAC), and five protocols: SAML, OAuth 2.0, OpenID Connect (OIDC), SCIM, and LDAP. For each, it covers what the standard does, the problem it solves, how it works, where it fits, and the security controls that matter. It closes with five cross-industry use cases, including medical device manufacturing, healthcare, and financial technology, and a set of governance practices.

Key takeaways

- **Federation centralizes authentication.** Applications trust a single identity provider (IdP), so phishing-resistant MFA, conditional access, and termination apply everywhere at once instead of app by app.
- **SAML and OIDC carry identity; OAuth carries permission.** SAML and OIDC tell an application who the user is. OAuth 2.0 gives a client a scoped, time-limited token to call an API. Confusing the two is a common design flaw.
- **SCIM and LDAP manage the lifecycle.** LDAP is the long-standing protocol for directories; SCIM is the modern HTTP API that provisions and deprovisions accounts in cloud applications.
- **RBAC and ABAC are complementary.** Roles give a governable baseline that auditors can certify; attributes add context, such as data sensitivity, location, device health, or relationship, where roles alone would explode in number.

Design principle: authenticate once, strongly, at a governed identity provider; provision and remove access automatically from an authoritative source; and authorize every request with the least privilege the task requires.

How the pieces fit together

The standards are easiest to understand as three layers. The lifecycle layer creates and removes identities. The authentication layer proves who is present and federates that proof to applications. The authorization layer decides what an authenticated user, service, or agent may do. Figure 1 shows each layer and the standards that operate in it.

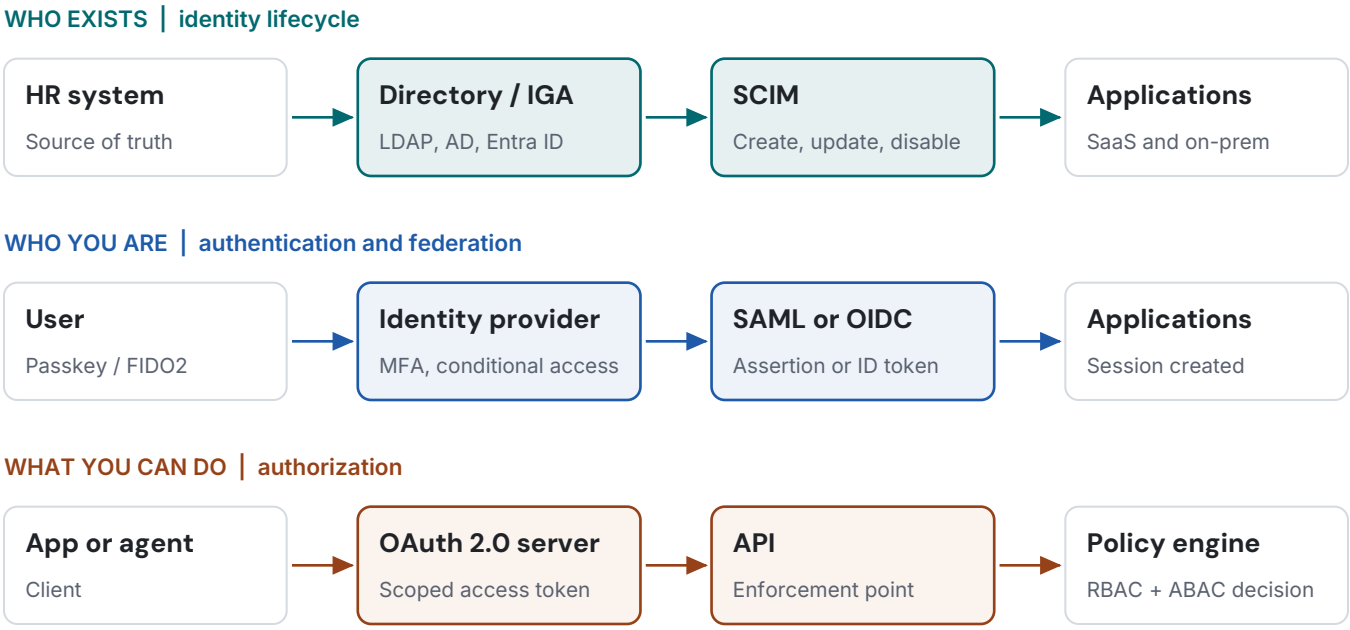


Figure 1. The identity fabric: lifecycle, authentication, and authorization layers

Standard or model	Primary job	Answers the question	Typical format and transport
LDAP	Directory access	Who exists, and what groups are they in?	Binary protocol (ASN.1) over TCP 389 / LDAPS 636
SCIM 2.0	Provisioning	Does this account exist in each app, with the right attributes?	JSON over HTTPS REST
SAML 2.0	Federated authentication (SSO)	Who is this user, as vouched for by the IdP?	Signed XML assertions via browser redirects and POSTs
OpenID Connect	Federated authentication (SSO)	Who is this user, for web, mobile, and API-era apps?	Signed JSON Web Tokens (ID tokens) over OAuth 2.0
OAuth 2.0	Delegated authorization	What may this client do on the user's behalf, or its own?	Access tokens (often JWTs) over HTTPS
RBAC	Access model	What does this job function permit?	Roles, permissions, role hierarchies, SoD constraints
ABAC	Access model	Is this request allowed given its full context?	Policies evaluated over subject, resource, action, environment attributes

Federation

What it is. Identity federation is a trust relationship in which one party, the identity provider, authenticates a user and another party, the service provider or relying party, accepts that result without holding the user's password. The trust is established ahead of time by exchanging metadata: endpoints, identifiers, and the public keys used to verify signed assertions or tokens. SAML and OIDC are the two dominant federation protocols.

Problems it solves

- **Password sprawl.** Without federation, each application stores its own credentials, and users reuse passwords across them. Federation removes the password from the application entirely.
- **Inconsistent authentication strength.** MFA configured separately in 200 applications will be uneven. With federation, one IdP enforces phishing-resistant MFA, device trust, and risk-based conditional access for all connected applications.
- **Slow, incomplete termination.** Disabling one IdP account ends new sign-ins everywhere. Paired with SCIM deprovisioning and session revocation, it closes the gap that orphaned accounts create.
- **Cross-organization access.** Partners, customers, contractors, and acquired companies can use their own identity provider while the resource owner keeps control of authorization.

How it is used

Pattern	Description	Example
Workforce SSO	Employees sign in once to a corporate IdP and reach SaaS and internal applications.	Entra ID or Okta federating to Salesforce, ServiceNow, Workday, and SAP
B2B federation	A partner's IdP is trusted for its own users; the host keeps authorization and access reviews.	Distributor or hospital staff reaching a manufacturer's partner portal
Customer identity (CIAM)	Consumers sign in with a hosted IdP, social login, or passkeys.	Banking app, patient portal, or retail account
Hub-and-spoke broker	One broker IdP federates with many upstream IdPs and presents a single trust point to applications.	Post-merger integration of multiple directories
Workload federation	Workloads exchange a platform-issued token for cloud credentials, removing long-lived secrets.	CI/CD pipeline assuming a cloud role with an OIDC token

Security considerations

- **The IdP becomes tier-zero.** Compromise of the IdP or its token-signing key lets an attacker forge assertions for every connected application. Protect signing keys in an HSM or managed key service, restrict administrative roles with privileged access controls, and monitor for anomalous token issuance.
- **Assurance level must match risk.** NIST SP 800-63-4 defines authentication and federation assurance levels and treats phishing resistance as a core requirement at higher levels.[1] Passwords and look-up secrets are not phishing-resistant.[2]

- **Validate everything.** Relying parties must check signature, issuer, audience, validity window, and replay protections on every assertion or token.
- **Govern the trust inventory.** Keep a register of every federation trust, its owner, certificate expiry, and attribute release, and recertify it.

1. NIST, SP 800-63-4 Digital Identity Guidelines, https://pages.nist.gov/800-63-4/sp800-63.html
2. NIST, SP 800-63B-4 Authenticators, https://pages.nist.gov/800-63-4/sp800-63b/authenticators/

RBAC and ABAC

Federation and protocols establish who is making a request. Access control models decide whether the request should succeed. Most mature programs use both RBAC and ABAC.

Role-based access control (RBAC)

What it is. In RBAC, permissions are assigned to roles and users are assigned to roles. ANSI INCITS 359 defines a role as a job function within the organization with associated authority and responsibility.[9] The standard model includes core RBAC, role hierarchies, and constraints such as static and dynamic segregation of duties (SoD).

Problems it solves. RBAC turns thousands of individual grants into a manageable set of business-meaningful bundles. It makes birthright access automatic for joiners, lets movers change roles cleanly, gives auditors something certifiable, and allows SoD rules such as "no one may both create a vendor and approve its payment."

Limitations. Roles are static. When access depends on context, such as region, data classification, time, device posture, or a specific customer relationship, organizations create ever more narrowly scoped roles. This "role explosion" undermines the governance benefit RBAC was meant to deliver.

Attribute-based access control (ABAC)

What it is. NIST SP 800-162 defines ABAC as an access control method in which authorization is determined by evaluating attributes of the subject, the object, the requested operation, and in some cases the environment, against policy.[10] A policy decision point (PDP) evaluates the rules; policy enforcement points (PEPs) in gateways, applications, and APIs apply the result.

Problems it solves. ABAC expresses fine-grained, contextual rules without multiplying roles: "a clinician may view a patient record only if they are on that patient's care team," or "a payment above $10,000 requires a step-up authenticated session from a managed device." It suits APIs, multi-tenant SaaS, data platforms, and AI agents.

Dimension	RBAC	ABAC
Decision basis	Membership in a role	Attributes of subject, resource, action, and environment
Granularity	Coarse to medium	Fine, down to the individual record or transaction
Context awareness	Limited	Native: time, location, device, risk, relationship

Dimension	RBAC	ABAC
Governance and audit	Straightforward: certify roles and memberships	Harder: requires policy review, testing, and attribute quality
Main risk	Role explosion, privilege creep	Poor attribute quality, untested policy logic
Best fit	Workforce baseline, ERP, birthright access, SoD	APIs, data access, customer data, dynamic and agent-driven decisions

Using them together

- **Roles for the baseline, attributes for the boundary.** RBAC grants a teller or engineer their standard capabilities; ABAC narrows each request by branch, product line, data classification, or risk.
- **Carry attributes in tokens.** SAML attributes, OIDC claims, and OAuth scopes deliver role and attribute data from the IdP to the enforcement point.
- **Treat policy as code.** Engines such as Open Policy Agent or Amazon Verified Permissions (Cedar) keep policies version-controlled, tested, and reviewed like software.
- **Related models.** Relationship-based access control (ReBAC) evaluates graph relationships such as owner, team member, or delegate, and is common in collaboration and consumer apps.

9. ANSI, Role Based Access Control (RBAC), INCITS 359, https://blog.ansi.org/ansi/role-based-access-control-rbac-incits-359/

10. NIST, SP 800-162 Guide to Attribute Based Access Control, https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-162.pdf

SAML 2.0

At a glance	
Full name	Security Assertion Markup Language 2.0
Steward	OASIS; approved as an OASIS Standard in March 2005[3]
Purpose	Federated authentication and attribute exchange, mainly browser-based single sign-on
Key artifacts	Assertions, metadata, bindings (HTTP Redirect, HTTP POST, Artifact), profiles (Web Browser SSO, Single Logout)

What it does

SAML lets an IdP send a digitally signed XML assertion to a service provider (SP) stating that a user authenticated, when and how they did, and which attributes apply, such as employee ID, email, department, and group memberships. The SP trusts the assertion because it verifies the signature with the IdP's public key from metadata.

How it works

- The user opens the application (SP-initiated) or clicks a tile on the IdP portal (IdP-initiated).
- The SP redirects the browser to the IdP with an AuthnRequest.
- The IdP authenticates the user, ideally with a passkey or other phishing-resistant method, and evaluates conditional access policy.

- The IdP returns a signed, optionally encrypted assertion to the SP's Assertion Consumer Service URL through the browser.
- The SP validates the signature, issuer, audience, time window, and InResponseTo value, maps attributes to local roles, and creates a session.

Common use cases

- Workforce SSO into enterprise SaaS such as HR, ERP, CRM, and ITSM platforms.
- B2B partner access, where a customer or supplier IdP is trusted for its own users.
- Higher education and research federations that let institutions share resources.
- Legacy and packaged enterprise applications that support SAML but not OIDC.

Security considerations

- Require signed assertions and validate them strictly; XML signature wrapping flaws have historically allowed forged identities.
- Protect the IdP token-signing certificate as a tier-zero asset. Theft enables forged assertions, often called a "Golden SAML" attack.
- Prefer SP-initiated flows. Disable IdP-initiated SSO where it is not needed, because unsolicited responses are harder to bind to a request.
- Track certificate rollover dates and keep assertion lifetimes short.

3. OASIS, Security Assertion Markup Language (SAML) v2.0, https://www.oasis-open.org/standard/saml/

OAuth 2.0

At a glance	
Full name	The OAuth 2.0 Authorization Framework (RFC 6749) and related RFCs
Steward	IETF. RFC 9700 (January 2025) is the current security best practice.[4] OAuth 2.1, which consolidates it, remains an Internet-Draft as of September 2026.[5]
Purpose	Delegated authorization: letting a client call an API with limited, revocable permission
Key artifacts	Access tokens, refresh tokens, scopes, grant types, authorization server, resource server

What it does

OAuth lets a user grant an application limited access to their resources without sharing a password, and lets services obtain tokens for their own API calls. The authorization server issues an access token that states which client may do what, for how long, and on whose behalf. The API, the resource server, validates the token and enforces its scopes. **OAuth is not an authentication protocol**: an access token proves permission to call an API, not the identity of the person at the keyboard.

How it is used

Grant or extension	When to use it	Notes
Authorization code + PKCE	Web, mobile, and single-page apps acting for a user	Default for user-facing clients; PKCE prevents code interception
Client credentials	Service-to-service calls with no user present	Use private-key JWT or mTLS instead of shared secrets
Device authorization	TVs, CLIs, kiosks, and devices without a browser	User approves on a second device
Refresh tokens	Long sessions without repeated sign-in	Rotate on use and bind to the client
Token exchange (RFC 8693)	A service or AI agent needs a narrower token to call a downstream API for the user	Supports delegation and down-scoping[14]
DPoP or mTLS binding	High-value APIs	Sender-constrained tokens that are useless if stolen

Common use cases

- A budgeting app reading account data from a bank API with the customer's consent.
- A mobile app calling its own back-end APIs with short-lived, scoped tokens.
- Microservices and batch jobs authenticating to each other with client credentials instead of embedded passwords.
- AI agents and Model Context Protocol (MCP) servers calling tools with user-bound, down-scoped tokens rather than broad service credentials.

Security considerations

- RFC 9700 deprecates the implicit and resource owner password credentials grants and calls for PKCE, exact redirect URI matching, and sender-constrained tokens.[4]
- Keep access tokens short-lived, validate audience and issuer at every API, and never pass a user's token through unchanged to a downstream service.
- Design scopes around business actions, such as *accounts.read* or *payments.initiate*, not around endpoints.

4. IETF, RFC 9700 Best Current Practice for OAuth 2.0 Security, https://datatracker.ietf.org/doc/rfc9700/

5. IETF, The OAuth 2.1 Authorization Framework (draft-ietf-oauth-v2-1-16), https://www.ietf.org/archive/id/draft-ietf-oauth-v2-1-16.html

14. IETF, RFC 8693 OAuth 2.0 Token Exchange, https://datatracker.ietf.org/doc/html/rfc8693

OpenID Connect

At a glance	
Full name	OpenID Connect 1.0
Steward	OpenID Foundation; nine OIDC specifications were published as ISO/IEC standards in 2024, including ISO/IEC 26131 (Core)[6]
Purpose	Federated authentication built on OAuth 2.0 for web, mobile, and API-era applications
Key artifacts	ID token (signed JWT), UserInfo endpoint, discovery document, JWKS, standard claims

What it does

OIDC adds an identity layer to OAuth 2.0. Alongside the access token, the OpenID provider returns an ID token: a signed JSON Web Token stating who the user is (the *sub* claim), who issued the token, which client it is for, when and how the user authenticated, and optional profile claims. Discovery and published signing keys (JWKS) make integration largely self-configuring.

How it works

- The client redirects the user to the provider with *scope=openid*, a PKCE challenge, a state value, and a nonce.
- The provider authenticates the user and returns an authorization code.
- The client exchanges the code at the token endpoint for an ID token and, if requested, an access token and refresh token.
- The client validates the ID token's signature, issuer, audience, expiry, and nonce, and can check *acr* and *amr* claims to confirm the authentication strength.

SAML or OIDC?

Consideration	SAML 2.0	OpenID Connect
Data format	XML assertions	JSON Web Tokens
Best for	Browser SSO to enterprise and legacy apps	Modern web, mobile, SPA, and API-driven apps
API access	Not designed for it	Native, through the underlying OAuth tokens
Mobile and native apps	Awkward	Designed for them
Direction of travel	Stable, widely deployed	Default for new builds and customer identity

Common use cases

- "Sign in with" workforce or social identity for SaaS and consumer applications.
- Customer identity platforms for banking, retail, and patient portals, with passkeys as the primary authenticator.

- Healthcare apps launched from an electronic health record through SMART on FHIR, which profiles OAuth 2.0 and OIDC.[15]
- Workload identity federation, where CI/CD systems present OIDC tokens to cloud providers instead of storing static keys.

Security considerations

- Never use an access token as proof of identity at the client; use the validated ID token.
- Use the authorization code flow with PKCE, validate the nonce, and store tokens server-side (a backend-for-frontend) in browser applications.
- Request an *acr* value for high-risk actions and require step-up authentication when it is not met.

6. OpenID Foundation, OpenID Connect Published as ISO/IEC Specifications, https://openid.net/10-years-on-openidconnect-published-as-iso-spec/

15. SMART Health IT, SMART on FHIR Authorization Best Practices, https://docs.smarthealthit.org/authorization/best-practices/

SCIM 2.0

At a glance	
Full name	System for Cross-domain Identity Management 2.0
Steward	IETF; RFC 7642 (concepts), RFC 7643 (core schema), RFC 7644 (protocol), published September 2015[7]
Purpose	Standard REST API and schema for provisioning, updating, and deprovisioning users and groups across domains
Key artifacts	/Users and /Groups resources, enterprise user extension, PATCH operations, filtering, bulk requests

What it does

Federation controls sign-in, but many applications still need a local account with the correct attributes and entitlements. SCIM lets an identity provider or IGA platform create, update, and disable those accounts automatically through a common JSON-over-HTTPS API, instead of CSV uploads, custom connectors, or manual tickets.

Problems it solves

- **Orphaned accounts.** When an employee leaves, SCIM sets *active=false* or deletes the account in every connected app within minutes, rather than waiting for someone to file tickets.
- **Mover drift.** Department, manager, and group changes propagate automatically, so entitlements follow the person's current job.
- **Integration cost.** One standard replaces a custom connector per application.
- **Audit evidence.** Provisioning events become logged, timestamped records that support joiner-mover-leaver testing.

How it works

The IdP or IGA platform acts as the SCIM client, and the application exposes a SCIM service provider endpoint protected with OAuth. An HR event, such as a hire, a transfer, or a termination, triggers POST, PATCH, or DELETE calls against /Users and /Groups. Group membership pushed through SCIM typically maps to application roles, which connects provisioning to RBAC.

Common use cases

- Automated joiner-mover-leaver across SaaS such as Slack, Zoom, GitHub, Salesforce, and ServiceNow.
- Pushing group-based role assignments that the target application enforces.
- B2B tenant provisioning, where a customer's IdP provisions its own users into a vendor's SaaS platform.
- Provisioning contractor and partner identities with enforced end dates.

Security considerations

- Protect the SCIM endpoint with scoped OAuth tokens and restrict which attributes the client may write; SCIM write access is effectively account-creation authority.
- Reconcile regularly: detect accounts created outside SCIM and treat them as exceptions.
- Deprovision and revoke sessions together; disabling an account does not always end its active tokens.

7. IETF, RFC 7644 SCIM Protocol, https://datatracker.ietf.org/doc/html/rfc7644

LDAP

At a glance	
Full name	Lightweight Directory Access Protocol, version 3
Steward	IETF; RFC 4511 (June 2006) defines the protocol, within the RFC 4510 series[8]
Purpose	Querying and modifying hierarchical directory services that hold users, groups, devices, and attributes
Key artifacts	Distinguished names (DNs), object classes, bind, search, modify operations; common implementations include Active Directory and OpenLDAP

What it does

LDAP is the protocol applications use to read and write a directory: to look up a user, check group membership, or authenticate by binding with a DN and password. Directories organize entries in a tree, such as *cn=jdoe,ou=Engineering,dc=example,dc=com*, and many enterprise systems still depend on it.

Problems it solves

- A single authoritative store for identities and groups, rather than per-application user tables.
- Fast, standardized lookups for authorization decisions, email routing, and address books.

- Authentication for infrastructure, network devices, Linux hosts, and older applications that predate federation.

Common use cases

- Active Directory as the core workforce directory and the upstream source for cloud IdP synchronization.
- Network devices, VPNs, and Wi-Fi using LDAP groups, often with RADIUS, for administrator and user access.
- Manufacturing, OT, and laboratory systems that support only LDAP authentication.
- Legacy line-of-business applications that bind directly to the directory.

Security considerations

- Require LDAPS or StartTLS and LDAP signing and channel binding; simple binds over clear text expose passwords.
- LDAP binds are password-based and not phishing-resistant. Where possible, place LDAP-only apps behind a federation gateway or application proxy.
- Lock down anonymous binds and restrict service accounts that bind to the directory; they are prime targets for reconnaissance and lateral movement.
- Monitor for bulk enumeration queries, which often precede privilege-escalation attacks against Active Directory.

8. IETF, RFC 4511 LDAP: The Protocol, https://datatracker.ietf.org/doc/rfc4511/

Where phishing-resistant authentication fits

Federation is what makes passwordless authentication practical at enterprise scale. Once applications rely on the IdP through SAML or OIDC, the organization can replace passwords and one-time codes with FIDO2 passkeys, platform authenticators such as Windows Hello for Business, or PIV and smart cards in one place, and every federated application inherits the stronger method.

The federal Phishing-Resistant Authenticator Playbook calls for removing options that fail against phishing, push bombing, SIM swap, and adversary-in-the-middle one-time code interception.[18] NIST SP 800-63-4 adds new options for phishing-resistant authentication and updates its threat models accordingly.[1]

Layer	Phishing-resistant practice
IdP sign-in	Passkeys or FIDO2 security keys for all users; hardware-bound keys for administrators
Federation	Require and verify authentication-context claims (SAML AuthnContext, OIDC acr/amr) for sensitive apps
Privileged access	Separate admin identities, just-in-time elevation, phishing-resistant MFA at every elevation, session recording
APIs and services	Sender-constrained OAuth tokens (mTLS or DPoP) and workload identity instead of static secrets
Legacy (LDAP-only)	Front with an application proxy or retire; compensate with network isolation and monitoring

1. NIST, SP 800-63-4 Digital Identity Guidelines, https://pages.nist.gov/800-63-4/sp800-63.html

18. IDManagement.gov, Phishing-Resistant Authenticator Playbook, https://www.idmanagement.gov/playbooks/altauthn/

Use cases across industries

The following scenarios show how the standards combine in practice. Each lists the goal, the design, and the control that matters most.

1. Global medical device manufacturer WORKFORCE, ERP, OT

Goal. Give employees and contractors across corporate, R&D, manufacturing, and field-service functions the right access on day one, and remove it on the last day.

- **Lifecycle:** the HR system triggers the IGA platform, which writes to Active Directory (LDAP) and provisions SaaS applications through SCIM.
- **Authentication:** the cloud IdP federates to SaaS via SAML and OIDC with passkeys; plant-floor and lab systems that support only LDAP sit behind an application proxy in segmented OT zones.
- **Authorization:** RBAC business roles drive birthright access to ERP, quality, and product lifecycle systems, with SoD rules such as separating quality-record approval from authorship.
- **Regulatory tie-in:** FDA medical device cybersecurity expectations include authentication and authorization controls in the device and its supporting systems.[17]

Control that matters most: automated leaver deprovisioning across LDAP and SCIM targets, with quarterly role certifications and SoD violation reporting as audit evidence.

2. Hospital system clinical applications HEALTHCARE

Goal. Let clinicians launch third-party apps from the electronic health record and see only the patients they are treating.

- **Authentication:** clinicians sign in once to the EHR through the hospital IdP; badge-tap plus PIN or passkey supports shared workstations.
- **Authorization:** SMART on FHIR uses OAuth 2.0 and OIDC to give each app a scoped token for specific FHIR resources.[15]
- **ABAC:** policy checks the care-team relationship, department, and break-glass status before releasing records.
- **Regulatory tie-in:** HIPAA requires unique user identification, emergency access procedures, and audit controls.[16]

Control that matters most: relationship-based ABAC with audited break-glass access, so emergency access is possible but always reviewed.

3. FinTech open finance and account aggregation FINTECH

Goal. Let a customer connect a budgeting or lending app to their bank accounts without sharing credentials, with consent they can see and revoke.

- **Authentication:** the customer signs in at the bank's OIDC provider with a passkey; the third-party app never sees the password.
- **Authorization:** OAuth 2.0 with PKCE, pushed authorization requests, and sender-constrained tokens, following the FAPI 2.0 Security Profile, finalized in February 2025.[11] The Financial Data Exchange (FDX) API aligns its security with FAPI.[12]
- **Consent and ABAC:** scopes map to specific accounts and data clusters; policy enforces consent duration, account ownership, and transaction limits.
- **Regulatory context:** the CFPB's Section 1033 personal financial data rights rule was finalized in October 2024 but is enjoined and under reconsideration.[13] Well-built consent and token controls remain valuable either way.

Control that matters most: bank-side consent management that binds each token to the customer, the approved accounts, and an expiry, with one-click revocation and full audit logging.

4. SaaS provider serving enterprise customers B2B SAAS

Goal. Let each enterprise customer bring its own identity provider and manage its own users in a multi-tenant platform.

- **Federation:** per-tenant SAML or OIDC connections to each customer's IdP, with enforced SSO for that email domain.
- **Provisioning:** each customer's IdP provisions and deprovisions its users through the platform's SCIM endpoint, scoped to that tenant.
- **Authorization:** RBAC roles within each tenant, plus ABAC checks that the tenant ID in the token matches the resource.

Control that matters most: tenant isolation enforced at the API from token claims, never from client-supplied tenant or account identifiers.

5. AI agents and non-human identities CROSS-INDUSTRY

Goal. Allow AI assistants, automation, and services to call APIs without long-lived secrets or more access than the user they serve.

- **Workload identity:** services authenticate with platform-issued OIDC tokens or mTLS certificates rather than static keys.
- **Delegation:** an AI agent exchanges the user's token for a narrower, audience-restricted token using OAuth 2.0 Token Exchange (RFC 8693).[14]
- **Governance:** every agent and service account has a named owner, defined scopes, rotation, monitoring, and a retirement date, just like a human identity.

Control that matters most: user-bound, down-scoped tokens with no token passthrough, so an agent can never exceed the entitlements of the person it acts for.

11. OpenID Foundation, FAPI 2.0 Security Profile (Final), https://openid.net/specs/fapi-security-profile-2_0-final.html
12. Financial Data Exchange, FDX Security Specification Boosted with FAPI, https://financialdataexchange.org/fdx-feed/fdx-security-specification-boosted-with-fapi/
13. Open Banking Tracker, Section 1033 Status and Timeline, https://www.openbankingtracker.com/guides/section-1033-status
14. IETF, RFC 8693 OAuth 2.0 Token Exchange, https://datatracker.ietf.org/doc/html/rfc8693
15. SMART Health IT, SMART on FHIR Authorization Best Practices, https://docs.smarthealthit.org/authorization/best-practices/
16. 45 CFR 164.312, HIPAA Technical Safeguards, https://www.law.cornell.edu/cfr/text/45/164.312
17. U.S. FDA, Medical Device Cybersecurity, https://www.fda.gov/medical-devices/digital-health-center-excellence/cybersecurity

Governance practices for an identity program

Standards provide interoperability; governance makes them trustworthy over time. The practices below turn the architecture into measurable, auditable controls.

Practice	What good looks like	Evidence and metrics
Authoritative sources	HR drives workforce identities; a sponsor system drives contractors and partners	Percentage of identities with an authoritative source and owner
Joiner-mover-leaver	SCIM and IGA automate provisioning and deprovisioning	Median time to deprovision; orphaned accounts found in reconciliation
Federation coverage	All in-scope apps behind the IdP; no local passwords	Percentage of apps federated; count of local accounts
Phishing-resistant MFA	Passkeys for all users; hardware-bound keys for admins	Coverage by population; legacy MFA exceptions
Role and SoD governance	Business-owned role model with defined SoD rules	Role count trend; SoD violations and remediation time
Access certification	Risk-based reviews: privileged quarterly, standard semi-annually	Completion rate; revocation rate; rubber-stamp detection
Privileged access	Just-in-time elevation, vaulting, session recording	Standing privileged accounts; elevation approvals
Non-human and AI identities	Owner, scope, rotation, and expiry for every service account and agent	Percentage with an owner; secrets older than policy
Trust and key hygiene	Register of federation trusts, signing keys, and certificates	Upcoming expiries; unused trusts removed
Identity threat detection	Detections for token theft, MFA fatigue, and anomalous federation changes	Mean time to detect and contain identity incidents

Common pitfalls

- Using OAuth access tokens as proof of login, or accepting ID tokens as API credentials.
- Federating sign-in but not provisioning, so accounts persist after termination.
- Letting roles multiply until nobody can certify them, instead of adding ABAC for context.
- Treating service accounts and AI agents as infrastructure rather than identities with owners and reviews.
- Leaving LDAP simple binds in clear text or allowing legacy protocols to bypass conditional access.

Conclusion

Federation, RBAC and ABAC, and the SAML, OAuth 2.0, OIDC, SCIM, and LDAP standards are not competing choices. Each solves a different part of the same problem. LDAP and SCIM decide who exists. SAML and OIDC prove who is present through a single, strongly protected identity provider. OAuth 2.0, RBAC, and ABAC decide what each request may do. Designed together and governed with clear ownership and evidence, they let an organization adopt passwordless authentication, cut orphaned and excessive access, and extend the same discipline to machines and AI agents.

About the author

Mark Adams is a senior IT risk, cybersecurity, and GRC professional with more than 30 years in technology, spanning security architecture, identity and access governance, privileged access, cloud security, audit and assurance, and AI security governance.

This paper is for educational purposes. Product names are examples, not endorsements. Standards status is current as of September 2026.