EDCS Security and Standards Guide

Enterprise Directory and Configuration Store

November 2026 · Describes the product as of November 12, 2026. EDCS has no numbered release yet; every installation we deliver is identified by its build, which we give you with it.

How to read this guide

This guide is written for security reviewers and assessors. For each standard it says what EDCS implements and where it stops. It states facts about the product, not conclusions about any framework: whether a control is satisfied in a given deployment is the assessor's decision. RFC numbers refer to the Internet Engineering Task Force's Request for Comments standards documents; API means application programming interface.

Two facts apply throughout. First, everything described here is built and covered by automated tests in the product. Second, most security features are **off by default** and are switched on per organization (tenant) or per deployment; where a feature is on by default, the text says so.

1. Standards supported

OAuth 2.0 and OpenID Connect

EDCS is an OAuth 2.0 authorization server and OpenID Connect (OIDC) provider.

Area	What is implemented
Discovery and keys	OIDC discovery document; published JSON Web Key Set (JWKS). Tokens are JSON Web Tokens (JWT) signed with RS256 (RSA 2048-bit keys).
Grants	Authorization code with Proof Key for Code Exchange (PKCE, S256 only); refresh token; client credentials; password. The password grant can be refused per organization or per client, so it can be retired.
Codes and refresh tokens	Authorization codes are single use. Refresh tokens rotate on every use; reuse of a spent refresh token revokes its whole family.
Client authentication	Client secret (stored as a hash); `private_key_jwt` signed assertions (RFC 7523); mutual Transport Layer Security (mTLS) client certificates, both certificate-authority-issued and self-signed (RFC 8705). One method per client, enforced at every endpoint; a request that offers two methods is refused.

Area	What is implemented
Certificate-bound tokens	Tokens issued to an mTLS client carry the thumbprint of its certificate (RFC 8705 section 3).
Token services	Introspection (RFC 7662), revocation (RFC 7009), user information endpoint with per-application attribute release.
Sign-out	OIDC sign-out (end-session endpoint) and back-channel logout to registered applications; administrator revocation of a person's sessions.
Sessions	Access tokens live 15 minutes by default. The sign-in session has an absolute lifetime and an idle timeout that is **on by default** at 15 minutes, adjustable per organization and per application.
Upstream OIDC	EDCS can accept sign-in from a partner's OIDC provider: authorization code with PKCE, `state` and `nonce`, strict issuer matching, signature checks with the partner's published keys only, and the RFC 9207 issuer check on the callback.

Security Assertion Markup Language (SAML) 2.0

Role	What is implemented	Where it stops
Identity provider	A trust registry of partner service providers with the attributes each receives; signed responses (RSA-SHA256, 3072-bit keys per organization); optional encrypted assertions; signed metadata; single logout; replay protection on inbound messages; `RequestedAuthnContext` mapped to assurance levels.	
Service provider (sign-in through a partner)	Signed AuthnRequests; responses by HTTP-POST only; the assertion itself must be signed, with defenses against XML signature wrapping; issuer, audience, recipient, `InResponseTo` and expiry are checked. Accounts are linked by an administrator, never automatically by default.	Unsolicited (identity-provider-initiated) responses and encrypted inbound assertions are not supported.

SCIM 2.0

System for Cross-domain Identity Management (SCIM) 2.0 in both directions:

- **Inbound (server):** users and groups, create, replace, patch, filter and delete, so an authoritative source such as a human-resources system keeps the directory current. Every change is recorded in the audit trail.

- **Outbound (client):** per-application connectors push account and entitlement changes to downstream applications by SCIM, or by a signed notification for applications without SCIM. A durable queue, idempotency keys, ordered delivery per person, retries with backoff and a dead-letter list mean a late or replayed message can never re-grant revoked access.

FIDO2 / WebAuthn security keys and passkeys

Fast IDentity Online 2 (FIDO2) / Web Authentication (WebAuthn), off until enabled for a deployment:

- A security key or passkey as the second factor after the password, or, per organization, passwordless sign-in with discoverable credentials and user verification (a PIN or biometric on the key). Passwordless sign-in asks for no user name, so it does not reveal whether an account exists.
- An organization can require a phishing-resistant factor. Once an account has a key, emailed codes are no longer offered to it.
- Adding or removing a key needs step-up proof, and on accounts with a key, a sign-in with a key or smart card in the last 15 minutes, so a phished password and a stolen token together cannot swap the key.

PIV, CAC, PIV-I and derived PIV

Smart-card sign-in with Common Access Card (CAC) and Personal Identity Verification (PIV) cards, off until enabled:

- The certificate must be within its validity period, carry the right key usage and an extended key usage for client authentication or smart-card logon, and chain to a trust anchor you configure; the operating system's store is never consulted and nothing is downloaded to build the chain.
- **Revocation is checked for the card and every intermediate, and fails closed:** Online Certificate Status Protocol (OCSP) first, then the certificate revocation list (CRL). If status cannot be confirmed, the sign-in is refused.
- The card is mapped to exactly one directory account (by default by the Department of Defense (DoD) identifier or principal name in the certificate; on a DoD Common Access Card, the Electronic Data Interchange Personal Identifier). Per organization, a policy can accept other agencies' and partners' cards: PIV, PIV-Interoperable (PIV-I) and External Certification Authority (ECA) certificate policies, card registrations by card universally unique identifier (UUID) or FASC-N (Federal Agency Smart Credential Number) bound to the issuing authority, and further certificate policies you name.
- **Derived PIV** credentials (National Institute of Standards and Technology (NIST) Special Publication (SP) 800-157), for example on a mobile device, sign in over the same path and are rated by their own certificate policy, never as the card they were derived from.
- Every sign-in and every refusal is audited with the certificate's subject, serial number, issuer and thumbprint.

Where it stops: nothing in this area has been tested with a real card, a real phone or a real partner public key infrastructure. The tests use certificates built by a test certificate authority with the real identifiers and name forms. There is no built-in profile for National Security Systems, coalition or Combined Communications-Electronics Board public key infrastructures; their policies can be named by hand.

Other sign-in methods

Authenticator-app codes (time-based one-time passwords, TOTP, RFC 6238) with a replay guard; single-use backup codes; trusted devices each person can list and revoke; email sign-in links and codes (single use, hashed, expiring). Email codes do not count as a second factor for assurance purposes and are not phishing resistant.

A sign-in notice (notice and consent banner), off by default, can be shown before any credential is asked for; it must be acknowledged explicitly, and each acknowledgment is recorded with the person, the application and the version of the text.

2. Assurance levels (NIST SP 800-63B)

Every user token carries `amr` (the methods used, RFC 8176), `auth_time` and `acr`: the authenticator assurance level (AAL1, AAL2 or AAL3) that EDCS computes from the factors it verified, mapped to NIST SP 800-63B Revision 3. Whether a deployment meets a level is your assessor's decision: the level is a label based on the factors EDCS verified, not an assessed level, and AAL3 in particular also depends on things outside this computation, such as validated cryptographic modules and the authenticator and verifier requirements of SP 800-63B. Nothing a client sends can raise the level, and a token refresh never raises it.

Level	Reached by
AAL1	One factor: a password, an emailed code or link, or a certificate without a PIN policy.
AAL2	A password plus an authenticator-app code or backup code; a software key or synced passkey with the password or with user verification; a software derived PIV credential where the organization relies on its activation.
AAL3	A hardware key unlocked by a proven second factor: a smart card whose certificate asserts a configured PIN policy, a hardware derived PIV credential where the organization relies on its activation, or a WebAuthn hardware key (attestation verified, not synced) with the password or user verification.

Step-up. An application can ask for a minimum level (`acr_values`) or a recent sign-in (`max_age`); EDCS then asks again rather than reusing a weaker or older session, and issues no code below the level. SAML `RequestedAuthnContext` maps to the same levels. Client libraries can refuse tokens below a configured level.

Service-to-service tokens (client credentials) carry no assurance level. An administrator impersonation token carries the administrator's own level and a claim that marks it as impersonation.

3. Cryptography and key handling

FIPS mode relies on the platform's module

EDCS performs its cryptography through the operating system's cryptographic library. **EDCS itself claims no Federal Information Processing Standard (FIPS) 140-3 validation.** In FIPS mode it relies on the validated module of the platform it runs on; whether a given deployment uses a validated module depends on that operating system build and its certificate, which EDCS does not check.

FIPS mode is off by default and is switched on per service. With it on, every service, before it opens a port or touches its database:

1. logs a statement of that reliance;
2. checks that the platform's FIPS provider is loaded and enforcing, and that a non-approved digest is actually refused;
3. refuses to start if any check fails. A setting that is not clearly on or off is treated as on, so a typing error cannot leave a deployment that asked for FIPS running without it.

In FIPS mode, smart-card chains are limited to approved key sizes and curves with no SHA-1 signatures, and WebAuthn registration offers only approved algorithms. A test in the build fails if any source file names a non-approved algorithm outside a short list of reviewed uses.

Use	Algorithm
Token signing	RS256, RSA 2048-bit
SAML signing	RSA-SHA256, RSA 3072-bit
Vault secrets, key envelopes, key ring	AES-256-GCM (Advanced Encryption Standard, 256-bit key, Galois/Counter Mode)
Session and antiforgery cookie protection	AES-256-CBC with HMAC-SHA256
Audit chain and checkpoints	SHA-256 (Secure Hash Algorithm) and HMAC-SHA256 (hash-based message authentication code)
Authenticator-app codes	HMAC-SHA-1 per RFC 6238 (approved for HMAC under NIST SP 800-131A Rev. 2)

Not covered by EDCS FIPS mode: the directory server's own password hashing, the database server and the TLS terminator are separate products and must run on FIPS platforms of their own; the procedure that creates and unlocks the vault's key-encryption key was not reviewed for FIPS mode. The standard EDCS images are built on an operating-system base without a FIPS provider and refuse to start in FIPS mode by design; a FIPS deployment uses images rebuilt on a FIPS-capable base.

Hardware-backed signing keys (PKCS#11)

Off by default. With a Public-Key Cryptography Standards #11 (PKCS#11) token configured (a network or PCIe hardware security module (HSM), or a cloud HSM's PKCS#11 client), EDCS generates its token signing keys and per-organization SAML signing keys **on the token**, marked sensitive and not extractable, and signs there. The database holds only the public key.

- **No silent fallback.** If the token is unreachable or the PIN is wrong, signing requests fail and the service reports itself not ready; it never makes a software key instead. A rejected PIN is not retried until restart, so EDCS never contributes to locking the token's PIN.
- When the HSM is turned on, the software token-signing key is rotated out and its private key erased. SAML organizations move to HSM keys at their next rollover, so partners that refresh metadata are never surprised.
- Not covered: RSA-PSS (probabilistic signature scheme) and elliptic-curve (ECDSA) keys in the HSM; the vault's key-encryption key and the cookie key ring are not held in the HSM. The HSM's own validation is its vendor's.

Key rotation and key custody

- Token and SAML signing keys roll over with a **publish-ahead** window (the next key is published before it signs) and a **retire-after** window (the old key stays published, signing nothing, while what it signed is still in use). Scheduled rotation is off by default; manual rotation and a key-compromise path that revokes all of an organization's keys are available to administrators.
- **Vault envelope encryption:** each secret version has its own 256-bit data key, wrapped by a key-encryption key that is decrypted at start-up from a passphrase-protected copy and held only in memory. It does not protect against an attacker with administrative control of the running host.
- Session cookie keys can be shared across instances in the database, sealed per service.

---

4. Audit chain and SIEM export

What is recorded

Every token grant and failure; smart-card sign-ins and refusals; second-factor enrollment, removal and reset; security-key registration and removal; sign-in policy changes; signing-key rotation; SAML partner changes; directory binds; SCIM changes; governance actions, denials and sampled policy decisions; entitlement grants and revocations; every secret read, write, delete and list; every configuration write and delete. Each record holds what, when (UTC to the microsecond), where (client address, service, organization), who, and the outcome.

How tampering is detected

- Each audit record is linked to the previous record of the same table and organization by a SHA-256 hash. A verification check, available to administrators only and limited to their own organization, reports edited, deleted, missing and unchained records.
- Optional **keyed checkpoints** (HMAC-SHA256, a keyed hash rather than a digital signature), chained to each other and to the point where the chain began, catch a rewritten chain. Their keys come from configuration, not the database, and can be rotated with the old keys kept for verification only.
- A fault in the chain never blocks a sign-in: the record is saved, marked, logged as an error and reported by the verifier.

This is tamper **evidence**, not prevention. A database administrator can still change records, and is then detected; deleting the newest checkpoints, or everything, shows only against a copy held outside the database, which is what SIEM export is for.

SIEM export

Off by default. Audit records and checkpoints are sent to your security information and event management (SIEM) system as syslog (RFC 5424 over TLS 1.2 or later, RFC 5425 framing) in JSON or Common Event Format (CEF), or as JSON over HTTPS. Delivery is at least once, with a stored position and bounded retries; destinations can differ per organization; chosen fields (for example client addresses) can be withheld.

Not included: per-record signing with an asymmetric key (non-repudiation), and an archival or retention job. Retention is set in the database and the SIEM.

---

5. Hardening

- **Accounts and sign-in:** per-address rate limits and per-account lockout (5 failures by default) with a progressive delay, shared across instances when configured; generic sign-in errors; masked password and code fields; show-password controls that do not submit.
- **Web responses:** HTTP Strict Transport Security and other security headers; a content security policy on the sign-in pages; a host allowlist setting against host-header injection.
- **Outbound calls:** back-channel logout and provisioning refuse private and internal addresses unless explicitly allowed, checked at connection time.
- **Containers:** six of the seven images are minimal runtime images with no shell and no package manager; every image declares a non-root user and a shell-free health check. The production deployment files run each service with a read-only root filesystem, every Linux capability dropped except DAC_OVERRIDE, which the sign-on and vault services add back while they still run as root (see What is not finished), and privilege escalation blocked. A self-assessment against the Center for Internet Security (CIS) container benchmark and the container requirements of the government's security guides grades each item met, partial or not met, with reasons.
- **What is not finished:** two services (the sign-on service and the vault) still run as the root user in the shipped deployment files, with one capability added back, because they read host files owned by another user; four more run as root by default until a host passes a documented check. Container images are not yet signed, and signatures are not verified on pull.

---

6. Software bill of materials and vulnerability scanning

- A software bill of materials (SBOM) in CycloneDX format is produced for every container image.
- Dependencies (including transitive ones) and every image are scanned for known vulnerabilities. The scan fails on any high or critical finding, or an unscored one, unless it is accepted in a reviewed exception list in which every entry carries a reason, an approver and an expiry date no more than a year out; an expired exception fails the scan.
- The scan runs on demand today. The scheduled pipeline (every change, every review and weekly) is defined and is waiting for a build runner to be attached.

7. NIST SP 800-53 Rev. 5: how EDCS supports these controls

The table summarizes how EDCS supports selected controls in the Access Control (AC), Identification and Authentication (IA), Audit and Accountability (AU) and System and Communications Protection (SC) families. It is not a statement that any control is satisfied. Every control also has a part that belongs to your organization (policy, procedure, infrastructure), summarized in the last column.

Control	How EDCS supports it	What stays with your organization
AC-2 Account management	Directory accounts; SCIM provisioning in and out; entitlement record with actor, authority and reason; access requests with approval; attestation campaigns; orphan and inactive account reports.	Account types, approvers, conditions, the authoritative source.
AC-2(4) Automated audit actions	Account and entitlement changes are audited and hash-chained.	Reviewing the records.
AC-3 Access enforcement	Every API checks token scopes and resolves the organization from the token, failing closed; separate databases and directory branches per organization; a policy decision point for applications.	Applications must call the decision point; EDCS cannot enforce inside them.
AC-5 Separation of duties	Rules block conflicting grants; reviewers cannot attest their own access.	Defining the conflicting pairs; separating operator duties.
AC-6 Least privilege	Scopes from group membership, narrowed to what a client asks for; delegations time-bound and narrower than the delegator.	Assigning the administrator scope sparingly.
AC-7 Unsuccessful logon attempts	Rate caps and account lockout with progressive delay, shared across instances when configured.	Setting thresholds to policy.
AC-8 System use notification	Optional notice and consent before any credential, acknowledgment audited.	Writing the approved text and switching it on.
AC-11 / AC-12 Session lock and termination	Absolute session lifetime; idle timeout on by default; short-lived access tokens; sign-out across applications; session revocation.	Screen lock on end-user devices.
IA-2 and IA-2(1)(2) Identification, authentication, multi-factor	Hosted sign-in; authenticator codes; security keys and passkeys; smart cards; SAML and OIDC federation; assurance levels and step-up; per-organization second-factor policy.	Requiring a second factor, and a phishing-resistant one, by policy.

Control	How EDCS supports it	What stays with your organization
IA-2(8) Replay resistance	PKCE; single-use codes; one-time-code replay guard; WebAuthn; SAML message replay ledger.	
IA-2(12) and IA-5(2) PIV and public-key authentication	Chain validation to configured anchors, fail-closed revocation, identity mapping.	Providing the trust store and the TLS front end that requests client certificates.
IA-5 Authenticator management	Client secrets and refresh tokens stored as hashes; one-time-code secrets encrypted; signing keys rotated.	Password complexity, age and history in the directory server's policy.
IA-7 Cryptographic module authentication	FIPS mode refuses to start without the platform's FIPS provider.	A platform whose module holds a validation certificate.
IA-8 Non-organizational users	SAML identity provider for partners; sign-in through partner identity providers.	Partner agreements.
IA-11 Re-authentication	Step-up to a higher level or a fresh sign-in.	Deciding which operations require it.
AU-2, AU-3, AU-12 Event logging and content	Records generated at the point of action with what, when, where, who and outcome.	Choosing which events the SIEM alerts on.
AU-6 and AU-9(2) Review; store on a separate system	SIEM export of records and checkpoints; activity views in the console.	Operating the SIEM; review and correlation.
AU-8 Time stamps	UTC to the microsecond.	Clock synchronization (Network Time Protocol, NTP).
AU-9 Protection of audit information	Hash chain, HMAC-keyed checkpoints, verification limited to administrators.	Restricting database access; keeping the checkpoint key off the database host.
SC-5 Denial-of-service protection	Per-address and per-endpoint rate limits.	Network-level protection.
SC-8 Transmission confidentiality	Security headers; TLS-only SIEM export with certificate validation.	TLS termination, certificates, internal network encryption.
SC-12 / SC-13 Key management and cryptographic protection	Key rotation with published keys; HSM-held signing keys; envelope-encrypted vault; FIPS mode.	Key custody procedures; the validated module.
SC-23 Session authenticity	Protected, secure, same-site session cookies; PKCE; signed SAML responses.	
SC-28 Protection at rest	Envelope-encrypted secrets; encrypted one-time-code secrets and key rings.	Database and disk encryption for other tables, including audit tables.

Controls EDCS does not address today: device identification to the network (IA-3) and non-repudiation of audit records (AU-10). The configuration, contingency, incident response, maintenance, media, physical, planning, personnel, risk, acquisition, system integrity and supply chain families depend mostly on the deploying organization; the deployment guide describes what EDCS contributes to availability and recovery.

What is not included

- **No government authorization and no validation.** EDCS has no Federal Risk and Authorization Management Program (FedRAMP) authorization, no Department of Defense (DoD) Impact Level authorization at any level, and no Authority to Operate (ATO). It has no FIPS 140-3 validated cryptographic module of its own and has not been assessed by a third party.
- **No DoD enterprise-service connectors yet.**
- **No hosted software-as-a-service offering yet.**
- **Not yet running in our own production.** Running in our own production: OAuth 2.0 and OpenID Connect sign-on service; Two-factor sign-in; Magic-link sign-in; Short-lived tokens; Directory with an API; Envelope-encrypted secrets vault; Central application configuration and feature flags; Per-application service clients; Audit events; Admin console; Administrator sign-in as a user (Quick Login); Multiple organizations, kept apart. Built and tested, shown on a test installation only: Browser single sign-on with PKCE; CAC and PIV smart-card sign-in; SAML 2.0 federation; Security keys and passkeys; Assurance levels and step-up; Sign-in through partner organizations; Sessions, token services and sign-out; Sign-in notice and consent; SCIM 2.0 provisioning; Outbound provisioning; Attribute-based access rules; Service clients without shared secrets; Entitlement record with attestation and reattestation; Separation of duties; Access requests with approval; Access decisions for your applications; Tamper-evident audit trail and SIEM export; FIPS mode on the platform's module; Signing keys in a hardware security module; Signing-key rotation; Several instances within one site; Delegated administration. Most are off by default. Smart-card sign-in has not been tested with real cards.
- **No conformance conclusion.** The control summary describes how EDCS supports each control; the assessor decides whether it is met.

Contact

Questions about this guide, or a walkthrough on a test installation: curtilage.zone/contact.

Curtilage · Acknowledged Development Inc.