Flex HR

Security & Compliance White Paper / Livre blanc sécurité

Multi-tenant HRIS — Sensitive HR data

Document généré à partir de l'implémentation applicative (septembre 2026 — FR/EN rév. 10)

Langues disponibles / Available languages

Français

English

Español

Deutsch

Livre blanc sécurité & conformité

Plateforme SIRH multi-tenant — Flex HR

Destinataires : services informatiques, RSSI, DPO, auditeurs

Version document : août 2026 (rév. 10) — export HMAC par clé d'intégration ; provisionnement sans auto-join domaine ; périmètre HR documents ; grant SU vs deploy clarifiés

---

Sommaire

1. Introduction

Ce livre blanc décrit les mesures de sécurité, de gouvernance des accès et de protection des données implémentées dans l'application Flex HR (SIRH multi-organisations). Il s'adresse aux équipes IT chargées d'évaluer le sérieux de la plateforme pour des données RH sensibles.

Pack d'alignement ISO (complément) : catalogue complet de l'annexe A ISO/IEC 27001:2022 (93 contrôles — gouvernance, RH, physique, technique) + cartographie 27017/27018/27701/25010 (statuts Implémenté / Partiel / Plateforme / Client / Écart ; colonne ownership App/Entreprise/Mixte) — iso-alignment-fluxehra.html / PDF. *Ce n'est pas une revendication de certification ISO.*

Principe directeur : le navigateur est considéré comme non fiable. Les rôles, le périmètre organisationnel et les opérations sensibles sont toujours validés côté serveur (ou par les politiques PostgreSQL Row Level Security sous le JWT de l'utilisateur).

- Authentification : Supabase Auth (GoTrue), flux PKCE, jetons JWT.
- Isolation locataire : schéma partagé PostgreSQL avec `organization_id` et RLS.
- RBAC applicatif : rôles `users`, matrice `role_permissions`, périmètres HR géographiques.
- Journalisation : table `audit_logs` avec attribution réelle / proxy.
- Chiffrement applicatif des champs sensibles (AES-256-GCM, préfixe `ENC2::`).

↑ Sommaire · ↑ Languages

2. Architecture technique

Frontend : Next.js (App Router), déployé sur Vercel.

Backend applicatif : routes API Next.js (Node.js) agissant comme couche de contrôle avant les opérations privilégiées.

Données & identité : Supabase (PostgreSQL, Auth, Storage, Realtime).

Modèle multi-tenant : schéma PostgreSQL partagé ; chaque table métier porte un `organization_id` protégé par RLS. Les Super Admins gèrent les modèles globaux (`default_schemas`, `settings`) distincts des données par organisation.

Données temporelles (bitemporal) : historisation des données employé avec dates d'effet ; intégrité assurée par la logique applicative et les contraintes base.

↑ Sommaire · ↑ Languages

3. Hébergement & localisation des données

Supabase (base de données, authentification, stockage fichiers) :

- Héberge PostgreSQL, Auth (GoTrue), Storage (compatible S3) et Realtime.
- La **région du projet** (ex. Union européenne : Francfort, Paris, Stockholm, etc.) est choisie à la création du projet Supabase dans le tableau de bord Supabase — ce choix détermine le lieu de stockage principal des données.
- L'application consomme Supabase via `NEXT_PUBLIC_SUPABASE_URL` et la clé anonyme côté client ; la clé `service_role` reste strictement serveur.

Vercel (application web & API) :

- Héberge le frontend Next.js et les routes API serverless/edge.
- La région d'exécution des fonctions est configurable dans les paramètres du projet Vercel (recommandation : aligner sur la région Supabase pour la latence et la souveraineté).
- **Production :** Vercel `dub1` (Dublin), configuré dans `vercel.json`, aligné sur Supabase `eu-west-1` (Irlande).
- Les fichiers volumineux (documents RH) sont téléversés **directement vers Supabase Storage**, pas via le corps des requêtes Vercel (réduction de surface et de coût).

Recommandation IT : documenter dans votre registre de traitement la région Supabase effective et la configuration régionale Vercel choisie pour chaque environnement (production, préproduction).

↑ Sommaire · ↑ Languages

4. Authentification & sessions

- **Supabase Auth** : connexion e-mail/mot de passe et OAuth (selon configuration projet) ; flux PKCE pour le client SSR Next.js (`@supabase/ssr`).
- **Jetons :** les routes API acceptent `Authorization: Bearer <access_token>` ; validation serveur via `supabase.auth.getUser(token)` . Jeton invalide → HTTP 401.
- **Synchronisation profil :** après connexion, création/mise à jour de la ligne `users` via `/api/auth/sync-profile` (service role après validation Bearer). **Pas d'attribution automatique** d'organisation ou de rôle admin depuis le domaine e-mail ou `admin_emails` — rattachement via invitation Manage Roles, création utilisateur admin, ou SSO org (utilisateur déjà provisionné). Le client n'attribue jamais de rôle en cas d'échec de sync.
- **Cloudflare Turnstile** (optionnel) : protection anti-bot sur vérification e-mail (`/api/auth/check-email` , `verify-human` , mot de passe oublié). Connexion SU Admin (`/login`) : bascule globale dans System Setup ; connexion organisation (`/{orgSlug}`) : bascule par org dans IT Link (`lib/turnstile-settings.ts`).
- **Déconnexion pour inactivité :** délai configurable (1–60 min) au niveau global (SU Admin → Setup) ou par organisation ; appliqué côté client avec politique lue via `/api/auth/session-policy` .
- **Invitations / réinitialisation mot de passe :** liens à usage unique via `generateLink` ; page `/reset-password` gère PKCE `?code=` et jetons de hash. Onboarding admin org : Super Admin ajoute l'e-mail → envoie le mail de bienvenue via `POST /api/admin/send-email` (étape séparée). URLs de redirection Supabase : inclure `https://fluxehra.ai/**` .
- **SSO organisation** (optionnel) : OIDC par organisation ; secret d'état HMAC (`ORG_AUTH_STATE_SECRET`) ; redirections d'erreur via `buildAppRedirectUrl` (`lib/app-origin.ts`) — origine canonique `APP_URL` (allowlist), codes d'erreur OIDC figés (pas de chaînes contrôlées par l'IdP dans l'URL) ; liens magic-link Supabase validés par hôte (`isSupabaseAuthRedirectUrl`).
- **DocuSign** (optionnel) : signature électronique des contrats de travail. Clé d'intégration + clé privée RSA au niveau plateforme (SU Admin → Setup, clé privée chiffrée) ; chaque organisation fournit Account ID + API User ID (IT Link) et consentement JWT one-shot (`/api/auth/docusign/callback` , `state` signé). Webhooks Connect (`/api/integrations/docusign/webhook`) : HMAC obligatoire (`DOCUSIGN_CONNECT_HMAC_SECRET`). Facturation DocuSign sur le compte client.
- **Demande de données employé (nouvelles embauches)** : flux public tokenisé pour compléter Employee Data / Local Data avant embauche. HR déclenche depuis New Hire (prérequis : prénom, nom, `personal_email`) ; lien unique `/employee-data-request/{token}` (90 jours). Jeton stocké en SHA-256 avec `EMPLOYEE_DATA_REQUEST_TOKEN_PEPPER` ; APIs publiques `/api/public/employee-data-request/*` (rate limit `public_auth` , Turnstile optionnel) ; champs sensibles chiffrés à l'enregistrement ; documents via URL signée sous `org/{orgId}/…` .
- **MFA TOTP (admin / super_admin / hr / Domain Admin) :** deux étapes — (1) inscription App Authenticator sur `/security/mfa` ; (2) **step-up à la connexion** (code TOTP après mot de passe) pour JWT `aal2` . **Par défaut ON en production** lorsque

`REQUIRE_MFA_FOR_PRIVILEGED_ROLES` est non défini ; `false` / `0` / `no` pour désactiver. Les APIs protégées refusent sans `aal2` → HTTP 403 `MFA_REQUIRED` (`lib/auth-mfa-policy.ts` , `lib/privileged-roles.ts` , `lib/privileged-mfa-access.ts` pour Domain Admin / HR via `role_permissions`). Activer TOTP Enroll + Verify dans Supabase Dashboard → Authentication → MFA.

- **Politique mot de passe :** page `/reset-password` exige ≥ 20 caractères avec majuscule, minuscule, chiffre et caractère spécial (`lib/password-policy.ts`) — aligner les réglages Auth Supabase Dashboard.
- **E-mails d'invitation HTML :** rendu via `lib/html-escape.ts` (sans jsdom/DOMPurify) pour compatibilité serverless Vercel — `lib/email-invitation-html.ts` n'importe pas `notification-html.ts` .

↑ Sommaire · ↑ Languages

5. Rôles & contrôle d'accès (RBAC)

La plateforme combine un rôle plateforme (`users.role`) et des permissions fines par organisation (`role_permissions`).

Rôle / profil	Description
Super Admin (`super_admin`)	Administration plateforme (schémas par défaut, déploiement config vers orgs) ; accès aux **données** d'une organisation uniquement après accord Org Admin (grant 48 h) puis entrée explicite. Le déploiement de champs/config par défaut ne requiert pas le grant (pas de lecture de dossiers RH).
Org Admin (`admin`)	Administration d'une organisation : schémas, paramètres, intégrations, mass upload.
Domain Admin	Accès lecture/écriture par module (Core HR, Performance, etc.) via `role_permissions` (`roleType = Domain Admin`).
HR	Périmètre géographique/organisationnel optionnel ; accès HR Self Service et écrans métier selon modules.
Manager	Dérivé de la chaîne hiérarchique (`manager` dans les données organisation) ; Manager Self Service.
Employé (`employee`)	Self-service employé ; visibilité des champs selon la matrice Data Visibility.

Matrice API serveur (`lib/supabase-api-auth.ts`) : `requireBearerDbUser` , `requireSuperAdminBearer` , `requireOrgAdminBearer` , `assertOrganizationScope` empêchent un admin d'org d'agir sur un autre `organization_id` .

Garde middleware (`proxy.ts`) : pages `/su-admin/*` réservées aux Super Admins ; **API en refus par défaut** — toute route `/api/*` exige une session valide sauf liste blanche explicite (`lib/middleware-routes.ts`) : auth publique, cron avec `CRON_SECRET` , intégrations HMAC.

↑ Sommaire · ↑ Languages

6. Isolation multi-tenant & Row Level Security

La frontière principale entre organisations est **PostgreSQL RLS** : les politiques lient `auth.uid()` au profil `users` et au `organization_id` des lignes.

- Le client navigateur utilise la clé anonyme Supabase + JWT utilisateur (jamais la `service_role`).
- La `service_role` est confinée aux routes serveur après contrôle d'autorisation (admin, cron, audit, crypto, sync profil).
- Modules Performance & Learning : RLS activé avec accès client refusé ; les APIs serveur appliquent les contrôles puis utilisent le service role.
- Table `proxy_sessions` : RLS sans accès direct client ; gestion via APIs dédiées.
- **Core HR (migration `20260606120000`) :** politiques tenant-scoped sur `employees`, `employee_historical_data`, `organizations` ; helpers `current_user_organization_id()` et `user_belongs_to_organization()` ; suppression des politiques permissives sur `schemas`.

↑ Sommaire · ↑ Languages

7. Proxy / impersonation (support & audit)

Fonctionnalité permettant à un administrateur de visualiser l'application comme un autre utilisateur (support, recette).

- **Intégrité de session :** le JWT Supabase identifie toujours l'utilisateur réel ; seul l'affichage applicatif (`appUser`) est surchargé.
- **Éligibilité :** validée côté serveur au démarrage du proxy (chaîne managériale, périmètre HR, règle « cross-admin strip »). Pour un Super Admin : grant Org Admin actif (48 h) et organisation entrée requise.
- **Revalidation continue :** chaque appel API protégé re-vérifie l'éligibilité via `assertProxySessionStillValid` (`lib/proxy-api-guard.ts`) ; session expirée ou règles violées → HTTP 403 et purge serveur.
- **Enregistrement serveur :** table `proxy_sessions`, TTL 12 heures, routes `/api/proxy/*`.
- **Audit :** les actions sont attribuées via l'enregistrement serveur `proxy_sessions` (et non un en-tête client falsifiable).
- **Non-autorisation :** le proxy ne confère pas de privilèges au-delà de ceux du JWT réel sous RLS.

↑ Sommaire · ↑ Languages

8. Protection des données & RGPD

L'application fournit des **mécanismes techniques** alignés sur les exigences RGPD ; la conformité juridique globale (base légale, registre, DPA avec sous-traitants) reste de la responsabilité du responsable de traitement.

- **Minimisation & visibilité :** matrice Data Visibility & Retention par champ (aucune / lecture / édition / workflow avec approbation HR).
- **Champs sensibles :** marquage `sensitive: true` dans les schémas ; chiffrement applicatif à l'enregistrement.
- **Rétention :** `retentionMonths` par champ (mois après départ) ; analyse et purge réelles via `/api/hr/retention` (`lib/retention-purge.ts`) — plus de comptages simulés.
- **DSAR :** export portable JSON et anonymisation in-place (employee HR Self Service) via `/api/hr/dsar/export` et `/api/hr/dsar/anonymize` ; blocage si legal hold org.
- **Trust Center :** page publique `/trust` (livres blancs, sous-traitants, contact DPA).
- **Documents :** types documentaires avec période de rétention et déclencheur d'expiration configurables.
- **Suppression organisation :** cascade documentée (`delete-organization-cascade`) incluant données org et journaux d'audit associés.
- **Sous-traitants :** Supabase, Vercel, Axiom (télémétrie de sécurité / SIEM — événements techniques sans contenu RH métier) — contrats et régions à documenter côté client.
- **Cookies marketing :** bandeau de consentement ; refus efface les cookies et déconnecte.

↑ Sommaire · ↑ Languages

9. Visibilité des champs & rétention

Onglet **Data Visibility & Retention** (SU Admin et configuration org) :

- Modes manager / employé : `none` , `view` , `edit` , `workflow` (demande de modification avec approbation HR).
- Champs sensibles (NSS, IBAN, etc.) : lecture manager par défaut, pas d'édition directe employé.
- Le marquage `sensitive: true` dans le schéma déclenche le **chiffrement en base** à l'enregistrement (voir section 10) — ce n'est pas un simple masquage d'interface.
- Données locales par pays : politique virtuelle unique « Any Local Data (Countries) » appliquée à tous les schémas `local-data-*` .
- Self-service : `self-service-submit-auth.ts` valide visibilité et rôle avant acceptation d'une demande de changement.

↑ Sommaire · ↑ Languages

10. Chiffrement — champs tagués sensibles en base de données

En transit :

- HTTPS obligatoire en production ; en-tête HSTS (`max-age=63072000`).
- Connexions Node → Supabase et SMTP : vérification TLS (`rejectUnauthorized: true`) sauf flags dev explicites interdits en production.

Au repos (infrastructure) :

- Chiffrement disque PostgreSQL et Storage : fourni par l'infrastructure Supabase (selon offre et région du projet).

Chiffrement applicatif effectif en base — champs `sensitive: true` :

Le drapeau `sensitive` sur un champ de schéma (ex. numéro de sécurité sociale, IBAN, permis de conduire, documents sensibles) n'est **pas** limité à l'interface. Lors de chaque enregistrement, la valeur en clair est transformée **avant insertion PostgreSQL** en texte chiffré préfixé `ENC2::` (**AES-256-GCM**, IV aléatoire, tag d'authentification, clé dérivée par scrypt de `ENCRYPTION_KEY`). Les valeurs historiques au format `ENC::` (AES CryptoJS) restent lisibles en rétrocompatibilité et seront ré-encryptées progressivement.

- **Table concernée :** colonne JSON `employee_historical_data.data` — les champs sensibles y sont stockés sous forme illisible sans la clé de déchiffrement.
- **Parcours d'écriture couverts :** embauche, mise à jour employé (HR Self Service, fiche employé), mass upload (`mass-upload-apply-server.ts`), génération de données de démo — via `updateEmployeeData` / `encryptValuesViaApi` → `/api/crypto/encrypt` .
- **Lecture :** le navigateur reçoit les valeurs chiffrées ; le déchiffrement pour affichage passe par `/api/crypto/decrypt` (Bearer JWT + périmètre organisation). Les rôles non privilégiés (employé, manager) ne peuvent déchiffrer que les valeurs présentes dans leur propre dossier ou leur arbre managérial — impossible de déchiffrer un texte chiffré arbitraire obtenu hors application.
- **Clé :** `ENCRYPTION_KEY` reste strictement côté serveur (jamais `NEXT_PUBLIC_*`). Sans cette clé, une lecture directe SQL des lignes `ENC2::…` / `ENC::…` ne restitue pas la donnée en clair.

Distinction importante pour les équipes IT :

- **Visibilité / rétention** (section 9) = qui peut voir ou modifier un champ dans l'application.
- **Sensible + chiffrement** = comment la valeur est *persistée* dans PostgreSQL une fois enregistrée.

Limites à connaître :

- Seuls les champs explicitement tagués `sensitive: true` dans le schéma sont chiffrés ainsi ; les autres champs restent en clair dans le JSON (protégés par RLS et RBAC).
- Les fichiers binaires (bucket Storage `documents` , privé) suivent un modèle d'accès distinct — route applicative authentifiée appliquant les règles de visibilité (section 13) — pas le préfixe `ENC::` .

- Le chiffrement est au niveau champ dans un document JSON, pas un chiffrement transparent de toute la base.

↑ Sommaire · ↑ Languages

11. Clés API & gestion des secrets

Secret	Usage	Exposition
`NEXT_PUBLIC_SUPABASE_ANON_KEY`	Client Supabase (soumise à RLS)	Publique (bundle client)
`SUPABASE_SERVICE_ROLE_KEY`	Opérations privilégiées serveur	Strictement serveur
`ENCRYPTION_KEY`	Chiffrement champs sensibles	Serveur uniquement
`CRON_SECRET`	Jobs planifiés (notifications, exports)	Serveur / planificateur
`INTEGRATION_KEY_PEPPER` / HMAC	Clés API intégration (hash SHA-256, jamais stockées en clair)	Serveur
`ORG_AUTH_STATE_SECRET`	SSO OIDC état signé	Serveur
`APP_URL`	Origine canonique pour redirections (anti open-redirect / Host spoofing)	Serveur (obligatoire en production)
`EMPLOYEE_DATA_REQUEST_TOKEN_PEPPER`	Hachage des jetons de demande de données employé (nouvelles embauches)	Serveur (obligatoire en production)
SMTP / Turnstile	E-mails et anti-bot	Table `settings` / `organization_settings`
`REQUIRE_MFA_FOR_PRIVILEGED_ROLES`	Exige JWT `aal2` pour admin / super_admin / hr / Domain Admin (défaut ON en production si non défini)	Vercel / env

Les paramètres globaux (`settings`) ne sont modifiables que par Super Admin. Régénération des clés d'intégration via API dédiée avec audit.

↑ Sommaire · ↑ Languages

12. Piste d’audit (audit trail)

Table `audit_logs` (PostgreSQL) :

- Champs : `organization_id` (nullable = action globale SU), `user_id`, `proxied_for_user_id`, `event_code`, `domain`, `employee_id`, `details`, `created_at`.
- Catalogue d’événements figé (`AUDIT_EVENTS`) : paramètres, schémas, organisations, mass upload, performance, learning, rôles, exports, `CORE_HR_HIRE` / `CORE_HR_UPDATE` / `CORE_HR_TERMINATE`, rétention, DSAR, preuves de conformité, vues de page, etc.
- Écriture serveur : `lib/audit-server.ts` ; client : `/api/audit/log` avec Bearer.
- UI : onglet Audit Trail (export CSV, export **Compliance evidence** JSON pour l’org, affichage proxy réel → cible).
- Vues de page : déduplication 1/min/route par session navigateur.

Les écritures Core HR embauche / mise à jour / sortie émettent désormais des événements d’audit (juillet 2026 rév. 4).

SIEM / détection d’intrusion (août 2026, rév. 6) : en complément de `audit_logs` (piste métier), l’application émet des événements de sécurité structurés `[SECURITY_EVENT]` via `lib/security-logger.ts` : 401/403, isolation tenant / BOLA, échecs d’auth, refus RLS, rate-limit, MFA, échecs cron, **HMAC webhook** (DocuSign Connect / export intégration), **blocages SSRF** (`ssrf-guard`). Ingest direct vers Axiom (dataset `fluxehra_logs`, hors Log Drain Vercel). Moniteurs e-mail (3 slots) : BOLA/tenant (403), force brute **par IP**, et signal fort (`CRON_AUTH_DENIED` / `RLS_DENIED` / `WEBHOOK_HMAC_DENIED` / `SSRF_BLOCKED`). Payloads sans PII (e-mail haché pour les échecs de login). Procédure interne : `docs/siem-security-operations.md` (non publique).

↑ Sommaire · ↑ Languages

13. Stockage documentaire (DMS)

- Bucket Supabase Storage `documents` **privé** (aucune lecture publique) ; chemins `org/{orgId}/employees/{employeeId}/{fieldId}/…`.
- **Lecture :** toutes les consultations passent par la route authentifiée `/api/documents/file/…` (`lib/document-access.ts`), qui applique les règles de l'application : visibilité du champ Employee/Manager (matrice Data Visibility), arbre managérial, périmètre HR/Admin par organisation, et inscription (enrollment) pour les contenus Learning/SCORM — chaque asset est autorisé individuellement puis servi en streaming (service role, support Range pour la vidéo).
- Téléversement : URL signée après `authorizeDocumentUpload` (contrôle rôle, périmètre org, arbre managérial) ; politique RLS storage limitant l'insert au dossier `org/{org de l'utilisateur}/…`.
- **Passerelle Storage** (`lib/storage-ops.ts`) : seul point d'accès au SDK Supabase Storage pour le bucket `documents` — chaque upload/download/remove/signed-url appelle `assertSafeStoragePath` avant l'opération.
- Protection chemins (`isUnsafeStoragePath`) : rejet de `..`, chemins absolus, backslashes, segments vides, encodage `%2e` (décodage multi-passes), caractères de contrôle ; `assertStoragePathUnderOrg` sur les APIs d'upload/delete (y compris flux public employee-data-request).
- Super Admin : accès global ; employé : ses propres documents selon la visibilité du champ ; manager : équipe ; HR/Admin : périmètre org ; photo (`employee_picture`) lisible par les membres de l'organisation (organigramme).

↑ Sommaire · ↑ Languages

14. Notifications & signalement d’incidents

- Notifications in-app : table dédiée + Supabase Realtime ; HTML sanitisé (DOMPurify, liste blanche de balises).
- E-mails : échappement HTML systématique sur modèles ; SMTP TLS vérifié en production.
- **Demande de données employé** : modèle `core-hr:employee-data-request` configurable (Core HR → Notifications) ; envoi via SMTP organisation après rotation du jeton.
- Cron notifications : `CRON_SECRET` + flag `NOTIFICATIONS_CRON_ENABLED`.
- **Bug reports manuels :** bouton « Report a bug » → ticket in-app escaladé HR → Org Admin → SU (widgets Dashboard) ; Clear / Forward ; pas d’e-mail sur ce chemin.
- **Bug reports automatiques :** erreurs client (`BugReportReporter` / ErrorBoundary) → e-mail SMTP vers `bugReportEmail` (défaut contact@fluxehra.ai) inchangé.
- **Masquage / fuite (ISO A.8.11) :** les chemins bug/erreur appliquent `redactSensitivePii` (IBAN, SSN, longues suites numériques) avant stockage ou e-mail. Les exports métier autorisés (CSV / API / SFTP / DSAR / UI) restent en clair après déchiffrement — le chiffrement `ENC2::` protège le repos, pas les canaux d’export justifiés.

↑ Sommaire · ↑ Languages

15. En-têtes HTTP & surface d'attaque

En-têtes de sécurité (`next.config.js` + `proxy.ts`) :

- `X-Frame-Options: DENY` , `X-Content-Type-Options: nosniff`
- `Strict-Transport-Security` (HSTS preload, 2 ans)
- **`Content-Security-Policy` à nonce par requête** (`lib/csp.ts` , appliquée dans `proxy.ts`) : en production, `script-src` n'autorise **ni** `'unsafe-inline'` **ni** `'unsafe-eval'` — chaque script inline est tagué d'un nonce aléatoire (128 bits) avec `'strict-dynamic'` ; les scripts injectés dynamiquement (Turnstile) héritent de la confiance. `style-src 'unsafe-inline'` est conservé comme risque résiduel documenté et accepté (Tailwind, Radix et bibliothèques de graphiques dépendent d'attributs de style inline).
- `X-Powered-By` supprimé (`poweredByHeader: false`) — pas de fuite d'information sur la pile technique.
- `Referrer-Policy: strict-origin-when-cross-origin`
- `Permissions-Policy` (caméra, micro, géolocalisation, paiement désactivés)

Gestion des dépendances (ISO A.8.8) : onglet SU Admin **Components Update** (dérive de versions vs registre npm) + SLA affiché (Critique ≤ 7 j, Élevé ≤ 30 j, Moyen ≤ 90 j). Gate CI `npm run audit:deps` (workflow GitHub `dependency-audit`) : échec sur avis **high/critical** des dépendances de production ; waivers documentés via `AUDIT_ALLOWLIST` .

Veille communautés (ISO A.5.6) : SU Admin **Security Watch** (CERT-FR / CISA / OSV, filtré impact) + Action GitHub quotidienne + Dependabot ; registre d'abonnements signé (2026-08-14).

Rate limiting global (`lib/api-rate-limit.ts` , table `api_rate_limits` en backend durable partagé entre instances, fallback mémoire) sur toutes les routes `/api/*` via le proxy :

- Authentifié : 600 req/min par utilisateur vérifié (uid JWT) et route, plus pré-filtre par IP
- Fichiers (`/api/documents/file/`) : 3 000 req/min par utilisateur (pages SCORM / documents multi-actifs)
- Auth publique (check-email, forgot-password, **employee-data-request**) : 30 req/min par IP
- Autres routes publiques : 120 req/min par IP
- Cron / intégrations (déjà protégés par secret) : 60 req/min par IP

Observabilité : `@vercel/analytics` et `@vercel/speed-insights` (métriques agrégées, sans données RH) ; CSP `connect-src` autorise `vitals.vercel-insights.com` . **SIEM Axiom** : ingest applicatif des événements `SECURITY_EVENT` (authz, tenant, RLS, brute-force par IP, cron, HMAC webhook, SSRF) vers `fluxehra_logs` ; alertes moniteur → e-mail ops. Événements ops structurés (`captureOpsEvent`) et webhook optionnel `OPS_ALERT_WEBHOOK_URL` .

Protection SSRF (ISO A.8.21) (`lib/ssrf-guard.ts`) : URLs/hôtes sortants configurables — webhooks d'export API, endpoints OIDC (token/JWKS + authorize), SSO SAML IdP, hôte SFTP, **et hôtes SMTP** (test + envoi) — https obligatoire pour les URLs IdP, résolution DNS, blocage des plages privées / loopback / lien-local / métadonnées cloud (RFC1918, 127.0.0.0/8, 169.254.0.0/16…). Validation à la sauvegarde Org SSO/OAuth et à l'authorize.

Autres contrôles : rate limiting export intégration (12 req/min), liste blanche IP optionnelle, **HMAC par clé d'intégration** (`Authorization: Bearer fx_…` + en-têtes signés avec la même clé — pas de secret HMAC plateforme), Turnstile anti-énumération e-mail (optionnel), réponses neutres anti-énumération sur les endpoints d'authentification publics, `robots.txt` restrictif, slugs organisation réservés (`lib/org-slug-policy.ts` — blocage `.env`, segments réservés), build TypeScript strict (sans `ignoreBuildErrors`). Lecteur SCORM : iframes `sandbox`, `postMessage` avec origine stricte, limites zip anti-bombe (100 Mo compressé / 500 Mo décompressé / 5 000 fichiers). **Documents RH :** HR à périmètre régional — même filtre sur la liste (`/api/documents/list`) et sur lecture directe (`/api/documents/file/…` via `employeeIdsInHrScope`).

Risques résiduels documentés : redirection locale marketing same-origin (`proxy.ts` — faux positif scanner) ; CSP absente sur HTML statiques publics (`iso-alignment-fluxehra.html` , docs Trust) — faible impact (pas de session app).

Régions production : Supabase **eu-west-1** (Irlande) et Vercel **dub1** (Dublin) — alignées dans `vercel.json` pour minimiser la latence et la résidence des données en Europe.

↑ Sommaire · ↑ Languages

16. Checklist de vérification IT

1. Confirmer RLS activé sur toutes les tables locataires dans Supabase (migration `20260606120000` pour Core HR).
2. Vérifier l'absence de `SUPABASE_TLS_INSECURE` et `SMTP_TLS_INSECURE` en production.
3. Régions documentées : Supabase `eu-west-1`, Vercel `dub1` (`vercel.json`) ; production `https://fluxehra.ai`.
4. Valider la force de `CRON_SECRET` et restreindre l'appelant du planificateur.
5. Activer Turnstile (SU global + org IT Link) si trafic bot suspect.
6. Tester tentatives IDOR (`organization_id` falsifié) sur APIs admin.
7. Surveiller le SIEM Axiom (`fluxehra_logs`) : moniteurs BOLA/tenant, brute-force par IP, high-signal (cron / RLS / HMAC / SSRF) ; plus rate-limit, échecs audit, `AUTH_UPSTREAM_UNAVAILABLE`, `403 MFA_REQUIRED`.
8. Confirmer `AXIOM_TOKEN` / `AXIOM_DATASET` / `AXIOM_URL` (edge) en production ; smoke ingest après rotation de token.
9. **MFA :** activer TOTP (Enroll + Verify) dans Supabase Dashboard → Authentication → MFA ; en production MFA est ON par défaut pour admin/super_admin/hr/Domain Admin (sinon définir explicitement `REQUIRE_MFA_FOR_PRIVILEGED_ROLES`) ; inscription sur `/security/mfa` **et** step-up TOTP à la connexion pour `aal2`.
10. Visiter `/trust` (Trust Center) ; télécharger le pack ISO `iso-alignment-fluxehra.pdf`.
11. Politique mot de passe : confirmer ≥ 20 + complexité côté app et dans Supabase Auth Dashboard.
12. URLs de redirection Supabase incluent `https://fluxehra.ai/**`.
13. Configurer `APP_URL=https://fluxehra.ai` et `EMPLOYEE_DATA_REQUEST_TOKEN_PEPPER` en production.
14. Configurer SSO OIDC par organisation (Org Setup) ou SAML au niveau projet Supabase.

↑ Sommaire · ↑ Languages

17. Écarts résiduels (transparence)

- Certification ISO / SOC : processus d'entreprise externe (pas une revendication produit) — voir pack d'alignement ISO.
- UX consentement ESS (`requiresConsent`) : `field_consents` + My preferences livrés.
- APM applicatif type Sentry toujours optionnel ; **SIEM sécurité Axiom** (rév. 6) : BOLA/tenant, force brute par IP, cron/RLS/HMAC/SSRF — pas un SOC 24×7 externalisé.
- Sauvegarde / PRA : engagement production Supabase Pro (backups quotidiens, rétention 7 j, RPO ≤ 24 h, RTO interne ≤ 4 h) — `docs/backup-restore.md` .
- **Écarts Annexe A honnêtes (rév. 9) :** DPA client (A.5.20 / A.5.31) ; revue indépendante du SMSI Foil Envie (A.5.35). A.5.6 clos (SU Admin Security Watch + abonnements signés 2026-08-14). A.8.23 clos (filtrage web N/A + compensations). Pack SMSI solo-opérateur signé 2026-08-13 (privé `Certification documentation/`).

Contrôles récemment renforcés (juin–juillet 2026) :

- En-têtes HTTP (CSP, HSTS, Permissions-Policy) et build TypeScript strict
- API en refus par défaut avec liste blanche explicite (`middleware-routes.ts`)
- Rate limiting global `/api/*` durable par utilisateur (table `api_rate_limits` + fallback mémoire)
- Revalidation proxy à chaque requête API protégée (`proxy-api-guard.ts`)
- Politiques RLS Core HR tenant-scoped (migration `20260606120000`)
- MFA TOTP pour admin / super_admin / hr / Domain Admin (défaut production ON), page `/security/mfa` , JWT `aal2`
- Correction IDOR : contrôle d'appartenance organisation dans la RPC `list_employee_directory` et guards de périmètre org sur les routes Performance / Learning
- Déchiffrement `/api/crypto/decrypt` restreint : rôles non privilégiés limités à leur dossier et arbre managérial
- Chiffrement applicatif AES-256-GCM (`ENC2::`) avec rétrocompatibilité `ENC::`
- Protection SSRF sur les URLs sortantes configurables (webhooks, OIDC, SFTP, SMTP, IdP authorize/SAML)
- Réponses neutres anti-énumération d'e-mails ; durcissement lecteur SCORM (sandbox, origines strictes, limites zip)
- Bucket `documents` passé en **privé** : lecture uniquement via la route authentifiée `/api/documents/file`
- SSO OIDC par organisation ; durcissement routage (`robots.txt` , slugs réservés)
- Régions alignées : Supabase eu-west-1 + Vercel dub1 ; production **https://fluxehra.ai**
- Turnstile : bascule globale SU Admin + bascule par organisation (IT Link)
- **CSP à nonce (rév. 2)** ; mises à jour dépendances ; garde Storage centralisée
- **Juillet 2026 (rév. 3)** : flux Employee data request ; anti open-redirect OIDC ; `lib/storage-ops.ts` ; Vercel Analytics

- **Août 2026 (rév. 5)** : SIEM sécurité Axiom (ingest direct `SECURITY_EVENT`, moniteurs BOLA/RLS/brute-force, procédure interne SIEM)
- **Août 2026 (rév. 6)** : instrumentation RLS/HMAC/SSRF ; moniteur high-signal ; brute-force par IP ; resserrage BOLA
- **Août 2026 (rév. 7)** : Annexe A A.8.8 / A.8.11 / A.8.21 Implémenté — `audit:deps` CI, rédaction PII bug/erreur, SSRF SMTP + IdP ; colonne ownership App/Entreprise/Mixte dans le pack ISO
- **Août 2026 (rév. 8)** : pack SMSI Foil Envie (gouvernance, RH, physique home-office, SDLC) **signé** — Annexe A ~75 Implemented ; Partial restants A.5.20/A.5.31/A.5.35
- **Août 2026 (rév. 9)** : A.5.6 Implemented (SU Admin Security Watch, Action quotidienne, Dependabot, registre d'abonnements signé) ; A.8.23 Implemented (filtrage N/A compensé) ; Annexe A **77 Implemented** / 3 Partial (A.5.20/31/35)
- **Août 2026 (rév. 10)** : export intégration HMAC signé par clé d'intégration (`fx_…`) ; fin auto-join domaine e-mail ; périmètre HR sur lecture document directe ; clarification grant SU (données) vs deploy (config plateforme)

Ce document reflète l'état de l'application au moment de sa génération. Pour toute question : contact@fluxehra.ai

↑ Sommaire · ↑ Languages

Security & Compliance White Paper

Multi-tenant HRIS Platform — Flex HR

Audience: IT departments, CISO, DPO, auditors

Document version: August 2026 (rev. 10) — per-integration export HMAC; no domain auto-join; HR document scope; SU grant vs deploy clarified

---

Table of contents

1. Introduction

This white paper describes security, access governance, and data protection measures implemented in the Flex HR application (multi-organization HRIS). It is intended for IT teams evaluating the platform for sensitive HR data.

ISO alignment pack (companion): full ISO/IEC 27001:2022 Annex A catalogue (93 controls — governance, HR, physical, technical) plus mapping to 27017, 27018, 27701, and 25010 (Implemented / Partial / Platform / Customer / Gap; App/Company/Mixed ownership column) — iso-alignment-fluxehra.html / PDF. *Not an ISO certification claim.*

Design principle: the browser is untrusted. Roles, organization scope, and sensitive operations are always enforced on the server (or by PostgreSQL Row Level Security under the user JWT).

- Authentication: Supabase Auth (GoTrue), PKCE flow, JWT tokens.
- Tenant isolation: shared PostgreSQL schema with `organization_id` and RLS.
- Application RBAC: `users` roles, `role_permissions` matrix, HR geographic scope.
- Logging: `audit_logs` table with real vs proxied identity.
- Application-level encryption for sensitive fields (AES-256-GCM, `ENC2::` prefix).

↑ Table of contents · ↑ Languages

2. Technical architecture

Frontend: Next.js (App Router), deployed on Vercel.

Application backend: Next.js API routes (Node.js) as a control layer before privileged operations.

Data & identity: Supabase (PostgreSQL, Auth, Storage, Realtime).

Multi-tenant model: shared PostgreSQL schema; tenant tables carry `organization_id` protected by RLS. Super Admins manage global templates separate from per-organization data.

Bitemporal data: employee history with effective dates; integrity via application logic and database constraints.

↑ Table of contents · ↑ Languages

3. Hosting & data location

Supabase (database, auth, file storage):

- Hosts PostgreSQL, Auth (GoTrue), Storage (S3-compatible), and Realtime.
- **Project region** (e.g. EU: Frankfurt, Paris, Stockholm) is selected when creating the Supabase project — this determines primary data location.
- The app uses `NEXT_PUBLIC_SUPABASE_URL` and the anon key on the client; `service_role` is server-only.

Vercel (web app & API):

- Hosts the Next.js frontend and serverless API routes.
- Function region is configurable in Vercel project settings (align with Supabase region when possible).
- Large HR documents upload **directly to Supabase Storage**, not through Vercel request bodies.

IT recommendation: record the effective Supabase region and Vercel configuration per environment in your processing register.

↑ Table of contents · ↑ Languages

4. Authentication & sessions

- **Supabase Auth**: email/password and OAuth (per project config); PKCE with Next.js SSR (`@supabase/ssr`). Production: **https://fluxehra.ai**.
- **Tokens:** APIs accept `Authorization: Bearer <access_token>` ; server validation via `getUser(token)` . Invalid token → 401.
- **Profile sync:** `/api/auth/sync-profile` after login (service role after Bearer check). **No automatic** org or admin role from email domain or `admin_emails` — assignment via Manage Roles invite, admin create-user, or org SSO (user pre-provisioned). Client never assigns roles on sync failure.
- **Cloudflare Turnstile** (optional): bot protection on check-email, verify-human, forgot-password. SU Admin login (`/login`): global toggle in System Setup; org sign-in (`/{orgSlug}`): per-org toggle in IT Link (`lib/turnstile-settings.ts`).
- **Idle logout:** configurable timeout (1–60 min) globally or per org via `/api/auth/session-policy` .
- **Invites / password reset:** single-use `generateLink` URLs; `/reset-password` handles PKCE `?code=` and hash tokens. Org admin onboarding: SU adds email → separate **Send welcome email** via `POST /api/admin/send-email` . Supabase redirect URLs must include `https://fluxehra.ai/**` .
- **Org SSO** (optional): per-organization OIDC with HMAC state secret. Error redirects use `buildAppRedirectUrl` (`lib/app-origin.ts`) — canonical `APP_URL` allowlist, fixed OIDC error codes only; Supabase magic links validated by host (`isSupabaseAuthRedirectUrl`).
- **DocuSign** (optional): employment contract e-signature. Platform Integration Key + encrypted RSA private key (SU Admin → Setup); each org supplies Account ID + API User ID (IT Link) and one-time JWT consent (`/api/auth/docusign/callback` , signed `state`). Connect webhooks (`/api/integrations/docusign/webhook`) require HMAC (`DOCUSIGN_CONNECT_HMAC_SECRET`). DocuSign billing remains on the customer account.
- **Employee data request (new hires)**: tokenized public flow for candidates to complete Employee/Local Data before hire. HR triggers from New Hire (requires first/last name + `personal_email`); link `/employee-data-request/{token}` (90-day expiry). Token stored as SHA-256 hash with `EMPLOYEE_DATA_REQUEST_TOKEN_PEPPER` ; public APIs `/api/public/employee-data-request/*` (rate limit `public_auth` , optional Turnstile); sensitive fields encrypted on save; documents via signed URLs under `org/{orgId}/…` .
- **MFA TOTP (admin / super_admin):** two steps — (1) enroll App Authenticator at `/security/mfa` ; (2) **login step-up** (TOTP after password) for JWT assurance `aal2` (`lib/mfa-step-up.ts` , `components/mfa-step-up-screen.tsx`). Separate from Vercel account 2FA. When `REQUIRE_MFA_FOR_PRIVILEGED_ROLES=true` (Vercel Production/Development), protected APIs require `aal2` or return 403 `MFA_REQUIRED` (`lib/auth-mfa-policy.ts`). Enable TOTP Enroll + Verify in Supabase Dashboard → Authentication → MFA.
- **Password policy:** `/reset-password` requires ≥ 20 characters with uppercase, lowercase, a digit and a special character (`lib/password-policy.ts`) — align Supabase Auth Dashboard settings.

- **Invite email HTML:** rendered via `lib/html-escape.ts` (no jsdom/DOMPurify) for Vercel serverless — `lib/email-invitation-html.ts` must not import `notification-html.ts`.

↑ Table of contents · ↑ Languages

5. Roles & access control (RBAC)

The platform combines platform role (`users.role`) and fine-grained org permissions (`role_permissions`).

Role	Description
Super Admin	Platform administration (default schemas, config deploy to orgs); **tenant data** only after Org Admin–approved 48h grant and explicit Enter. Default-field deploy does not require the grant (no HR record reads).
Org Admin	Single-organization administration.
Domain Admin	Per-module Read/Write via `role_permissions`.
HR	Optional geographic/org scope; HR Self Service.
Manager	Derived from management chain; Manager Self Service.
Employee	Employee self-service; field visibility per Data Visibility matrix.

Server API matrix (`supabase-api-auth.ts`) and middleware guards (`proxy.ts`) enforce scope. **Default-deny API:** all `/api/*` routes require a valid session except an explicit allowlist (`middleware-routes.ts`): public auth, cron with `CRON_SECRET`, HMAC integrations.

↑ Table of contents · ↑ Languages

6. Multi-tenant isolation & RLS

Primary boundary: **PostgreSQL RLS** linking `auth.uid()` to `users` and row `organization_id`.

- Browser uses anon key + user JWT (never `service_role`).
- `service_role` only in server routes after authorization.
- Performance/Learning: RLS deny client; server APIs enforce access.
- **Core HR (migration `20260606120000`):** tenant-scoped policies on `employees`, `employee_historical_data`, `organizations`; helpers `current_user_organization_id()`, `user_belongs_to_organization()`; permissive `schemas` policies removed.

↑ Table of contents · ↑ Languages

7. Proxy / impersonation

- Supabase JWT always identifies the real user; UI overlays target identity only.
- Eligibility validated server-side on proxy start. Super Admin requires an active Org Admin–approved 48h grant and must have entered that organization.
- **Continuous revalidation:** every protected API call re-checks eligibility via `assertProxySessionStillValid` (`proxy-api-guard.ts`); stale sessions cleared with HTTP 403.
- `proxy_sessions` table, 12h TTL.
- Audit uses server proxy record (not forgeable client headers).
- Proxy does not grant privileges beyond real user RLS.

↑ Table of contents · ↑ Languages

8. Data protection & GDPR

The application provides **technical mechanisms** aligned with GDPR expectations; legal compliance (lawful basis, register, DPAs) remains the controller's responsibility.

- Field visibility & retention matrix per field.
- Sensitive fields encrypted at save.
- `retentionMonths` per field; real analyze/purge via `/api/hr/retention` (`lib/retention-purge.ts`).
- DSAR portable JSON export and in-place anonymization (`/api/hr/dsar/*`); blocked when org legal hold is on.
- Public Trust Center at `/trust` (white papers, sub-processors, DPA contact).
- Document types with retention and expiration triggers.
- Organization deletion cascade includes org data and audit logs.
- Sub-processors: Supabase, Vercel, and Axiom (security SIEM telemetry — technical events, no HR business payloads) — document regions and contracts.
- Marketing cookie notice; refuse clears cookies and signs out.

↑ Table of contents · ↑ Languages

9. Field visibility & retention

Data Visibility & Retention tab: manager/employee modes `none`, `view`, `edit`, `workflow`. Sensitive fields (SSN, IBAN, etc.) restricted by default. The `sensitive: true` schema flag triggers **database encryption** on save (section 10) — not UI masking only. Virtual local-data policy for all `local-data-*` schemas. Self-service validated in `self-service-submit-auth.ts`.

↑ Table of contents · ↑ Languages

10. Encryption — sensitive-tagged fields in the database

In transit: HTTPS, HSTS, TLS verification for Supabase and SMTP.

At rest (infrastructure): Supabase disk/storage encryption per project region.

Effective application encryption — `sensitive: true` schema fields: The `sensitive` flag is not UI-only. On every save, plain values are encrypted **before PostgreSQL insert** as `ENC2::` + **AES-256-GCM** ciphertext (random IV, auth tag, scrypt-derived key from server-only `ENCRYPTION_KEY`). Legacy `ENC::` (CryptoJS AES) values stay readable for backward compatibility.

- **Storage:** `employee_historical_data.data` JSON column — sensitive field values persist as ciphertext.
- **Write paths:** hire, employee update, mass upload, demo seeding — via `updateEmployeeData` / `encryptValuesViaApi` → `/api/crypto/encrypt`.
- **Read:** decryption for display only via `/api/crypto/decrypt` (Bearer + org scope). Non-privileged roles (employee, manager) can only decrypt ciphertexts present in their own record or management subtree.
- **Visibility vs encryption:** Data Visibility controls who sees a field; `sensitive` controls how it is stored in the DB.

Limits: Only schema fields marked `sensitive: true` are encrypted this way; binary documents are protected by the private bucket + authenticated file route (section 13), not `ENC2::`.

↑ Table of contents · ↑ Languages

11. API keys & secrets

Anon key (client, RLS-bound); service role (server only); encryption key; cron secret; integration key hashing (SHA-256); OIDC state secret; `APP_URL` (canonical redirect origin, required in production); `EMPLOYEE_DATA_REQUEST_TOKEN_PEPPER` (new-hire data-request token hashing, required in production); SMTP/Turnstile in settings tables. Global settings: Super Admin only.

↑ Table of contents · ↑ Languages

12. Audit trail

`audit_logs` table with org scope, user, proxied user, event codes, details. Frozen `AUDIT_EVENTS` catalogue including `CORE_HR_HIRE` / `CORE_HR_UPDATE` / `CORE_HR_TERMINATE`, retention, DSAR, and compliance-evidence exports. CSV + compliance JSON export in Audit Trail UI. Page views deduped per minute.

Security SIEM (August 2026, rev. 6): alongside business `audit_logs`, the app emits structured `[SECURITY_EVENT]` telemetry via `lib/security-logger.ts` (401/403, tenant isolation / BOLA, auth failures, RLS denials, rate limits, cron auth failures, **webhook HMAC** denials for DocuSign Connect / integration export, **SSRF blocks**). Direct ingest to Axiom dataset `fluxehra_logs` (not Vercel Log Drain). Email monitors (3 slots): BOLA/tenant (403), brute-force **per IP**, and high-signal (`CRON_AUTH_DENIED` / `RLS_DENIED` / `WEBHOOK_HMAC_DENIED` / `SSRF_BLOCKED`). Payloads are PII-safe (hashed email on login failures). Internal ops procedure: `docs/siem-security-operations.md` (not public).

↑ Table of contents · ↑ Languages

13. Document storage (DMS)

Private Supabase bucket `documents` (no public read). All reads go through the authenticated app route `/api/documents/file/…` (`lib/document-access.ts`) enforcing application rules: Employee/Manager field visibility (Data Visibility matrix), management subtree, HR/Admin org scope, and learning enrollments for SCORM/course content — each asset individually authorized then streamed (service role, Range support for video). Uploads via signed URLs after authorization, org-scoped storage RLS insert policy. **Storage gateway** (`lib/storage-ops.ts`): sole entry point to the Supabase Storage SDK for `documents` — every operation calls `assertSafeStoragePath` first. Path hardening via `isUnsafeStoragePath` (traversal, encoded `%2e`, absolute paths, backslashes) plus `assertStoragePathUnderOrg` on upload/delete APIs (including public employee-data-request flow). `employee_picture` remains readable by org members (org chart).

↑ Table of contents · ↑ Languages

14. Notifications & incident reporting

Sanitized HTML notifications; TLS SMTP; cron secret. Configurable `core-hr:employee-data-request` email template for new-hire data requests. Manual « Report a bug » → in-app escalation HR → Org Admin → SU (no email). Automatic client-error reports still email configurable `bugReportEmail` . **ISO A.8.11:** bug/error channels run `redactSensitivePii` (IBAN/SSN-like patterns) before store/email; authorized CSV/API/SFTP/DSAR/UI decrypt remains plaintext for justified exports.

↑ Table of contents · ↑ Languages

15. HTTP headers & attack surface

Per-request nonce-based Content-Security-Policy (`lib/csp.ts` , applied in `proxy.ts`): in production, `script-src` allows **neither** `'unsafe-inline'` **nor** `'unsafe-eval'` — inline scripts carry a random 128-bit nonce with `'strict-dynamic'` ; dynamically injected scripts (Turnstile) inherit trust. `style-src` `'unsafe-inline'` is kept as a documented accepted risk (Tailwind/Radix/chart libraries rely on inline style attributes). `X-Powered-By` suppressed (`poweredByHeader: false`). Plus HSTS (2-year preload), X-Frame-Options DENY, nosniff, Permissions-Policy, Referrer-Policy.

Dependency management (ISO A.8.8): SU Admin **Components Update** (version drift vs npm) + in-UI SLA (Critical ≤7d, High ≤30d, Medium ≤90d). CI gate `npm run audit:deps` (GitHub `dependency-audit` workflow) fails on production **high/critical** advisories; documented waivers via `AUDIT_ALLOWLIST` .

Community subscriptions (ISO A.5.6): SU Admin **Security Watch** (CERT-FR / CISA / OSV, impact-filtered) + daily GitHub Action + Dependabot; signed subscriptions register (2026-08-14).

Global API rate limits (durable `api_rate_limits` table shared across instances, memory fallback): authenticated 600/min per verified user; file serving `/api/documents/file/` 3,000/min per user; public auth 30/min (includes `/api/public/employee-data-request`); other public 120/min; cron 60/min.

Vercel observability: `@vercel/analytics` and `@vercel/speed-insights` (aggregated metrics, no HR payload); CSP `connect-src` allows `vitals.vercel-insights.com` . **Axiom security SIEM:** application ingest of `SECURITY_EVENT` telemetry (authz, tenant, RLS, per-IP brute-force, cron, webhook HMAC, SSRF) to dataset `fluxehra_logs` ; monitor alerts to ops email. Structured ops events (`captureOpsEvent`) and optional `OPS_ALERT_WEBHOOK_URL` .

SSRF guard (ISO A.8.21) (`ssrf-guard.ts`) on tenant-configurable outbound destinations: export webhooks, OIDC token/JWKS + authorize URLs, SAML IdP SSO URL, SFTP hosts, **and SMTP hosts** (test + send) — https required for IdP URLs, DNS resolution, private/loopback/metadata ranges blocked; validated on Org SSO/OAuth save and authorize. Integration export limits, optional IP allow-list, HMAC exports, Turnstile anti-enumeration, neutral anti-enumeration responses on public auth endpoints, restrictive `robots.txt` , reserved org slugs (`org-slug-policy.ts`), strict TypeScript build. SCORM player: sandboxed iframes, strict postMessage origins, zip-bomb limits. Regions: Supabase eu-west-1 + Vercel dub1 (`vercel.json`).

↑ Table of contents · ↑ Languages

16. IT verification checklist

1. Confirm RLS on tenant tables (Core HR migration `20260606120000`).
2. No insecure TLS flags in production.
3. Document Supabase eu-west-1 and Vercel dub1 regions; production URL `https://fluxehra.ai`.
4. Set `APP_URL` and `EMPLOYEE_DATA_REQUEST_TOKEN_PEPPER` in production.
5. Strong CRON_SECRET.
6. Enable Turnstile globally (SU login) and/or per org (IT Link) if needed.
7. Test IDOR on admin APIs.
8. Monitor Axiom SIEM (`fluxehra_logs`): BOLA/tenant, per-IP brute-force, high-signal (cron / RLS / HMAC / SSRF); plus rate limits, audit failures, auth upstream errors, 403 MFA_REQUIRED.
9. Confirm `AXIOM_TOKEN` / `AXIOM_DATASET` / `AXIOM_URL` (edge) in production; re-run ingest smoke after token rotation.
10. **MFA:** enable TOTP Enroll+Verify in Supabase Dashboard; MFA defaults ON in production for admin/super_admin/hr (set `REQUIRE_MFA_FOR_PRIVILEGED_ROLES=false` only to disable); enroll at `/security/mfa` and complete TOTP step-up for `aal2`.
11. Password policy: app enforces ≥20 + complexity on `/reset-password`; align Supabase Auth dashboard.
12. Review `/trust` and download `iso-alignment-fluxehra.pdf`.
13. Supabase redirect URLs include `https://fluxehra.ai/**`.
14. Configure per-org OIDC SSO or project-level SAML.

↑ Table of contents · ↑ Languages

17. Residual gaps (transparency)

- ISO/SOC *certification* remains an external company process — see the ISO Alignment Pack (not a product claim).
- ESS consent UX for `requiresConsent` fields: `field_consents` + My preferences shipped.
- Optional Sentry APM; **Axiom security SIEM** (rev. 6) covers BOLA/tenant, per-IP brute-force, cron/RLS/HMAC/SSRF — not an outsourced 24×7 SOC.
- Backup/DR: Supabase Pro daily backups, 7-day retention, RPO ≤ 24h, internal RTO ≤ 4h — see `docs/backup-restore.md`.
- **Honest Annex A residuals (rev. 9):** customer DPA (A.5.20/A.5.31), independent Foil Envie ISMS review (A.5.35). A.5.6 closed (SU Admin Security Watch + signed subscriptions 2026-08-14). A.8.23 closed (web-filter N/A + compensating). Solo-operator ISMS pack signed 2026-08-13 (private `Certification documentation/`).

Recently strengthened (June–July 2026): HTTP security headers; default-deny API allowlist; durable rate limiting; proxy revalidation; Core HR tenant RLS; MFA TOTP for admin/SU/hr/Domain Admin with login step-up (default ON in production); IDOR fixes; scoped crypto decrypt; AES-256-GCM (`ENC2::`); SSRF guard; private documents bucket; routing hardening; eu-west-1 + dub1; production **https://fluxehra.ai**; Turnstile; nonce CSP (rev. 2); Storage path hardening; **July 2026 rev. 3:** Employee data request flow; OIDC open-redirect mitigation; `storage-ops.ts` ; Vercel Analytics. **July 2026 rev. 4:** Core HR hire/update/terminate audit; real retention purge; DSAR export/anonymize; MFA default-on + HR role; password policy ≥20; Trust Center `/trust` ; compliance evidence export; `npm run test:security-smoke` ; ISO alignment pack updated. **August 2026 rev. 5:** Axiom security SIEM (direct `SECURITY_EVENT` ingest, BOLA/RLS/brute-force monitors, internal SIEM procedure). **August 2026 rev. 6:** RLS/HMAC/SSRF instrumentation; high-signal monitor; per-IP brute-force; BOLA monitor tuning. **August 2026 Wave 1:** MFA for Domain Admin; IDOR guard tests; Pro backup commitment documented; solo IR attestation. **August 2026 rev. 7:** Annex A A.8.8 / A.8.11 / A.8.21 Implemented — `audit:deps` CI, bug/error PII redaction, SMTP+IdP SSRF; ISO pack ownership column (App/Company/Mixed). **August 2026 rev. 8:** Foil Envie ISMS maximize pack signed; Annex A ~75 Implemented; security-tests CI + non-prod URL guard. **August 2026 rev. 9:** A.5.6 Implemented (SU Admin Security Watch, daily Action, Dependabot, signed subscriptions); A.8.23 Implemented (web-filter N/A + compensating); Annex A **77 Implemented** / 3 Partial (A.5.20/31/35).

Contact: contact@fluxehra.ai

↑ Table of contents · ↑ Languages

Libro blanco de seguridad y cumplimiento

Plataforma SIRH multi-inquilino — Flex HR

Destinatarios: departamentos de TI, CISO, DPO, auditores

Versión del documento: agosto de 2026 (rev. 10) — HMAC de exportación por clave de integración; sin auto-join por dominio; alcance HR en documentos; grant SU vs deploy aclarados

---

Índice

1. Introducción

Este libro blanco describe las medidas de seguridad, gobernanza de accesos y protección de datos de Flex HR (SIRH multi-organización), dirigido a equipos de TI que evalúan datos RH sensibles.

Pack de alineación ISO (complemento, EN/FR): catálogo completo del anexo A ISO/IEC 27001:2022 (93 controles) — iso-alignment-fluxehra.html — no es una reclamación de certificación ISO.

Principio: el navegador no es de confianza; roles y operaciones sensibles se validan en el servidor o mediante RLS en PostgreSQL.

- Autenticación Supabase Auth (PKCE, JWT).
- Aislamiento por `organization_id` y RLS.
- RBAC con `users` y `role_permissions`.
- Auditoría en `audit_logs`.
- Cifrado de campos sensibles (`ENC2::` , AES-256-GCM).

↑ Índice · ↑ Languages

2. Arquitectura técnica

Frontend: Next.js en Vercel. **Backend:** rutas API Next.js. **Datos:** Supabase (PostgreSQL, Auth, Storage). Modelo multi-inquilino con RLS. Datos bitemporales para historial de empleados.

↑ Índice · ↑ Languages

3. Alojamiento y ubicación de datos

Supabase: región del proyecto elegida al crearlo (p. ej. UE: Fráncfort, París) — determina ubicación principal. Clave anónima en cliente; `service_role` solo servidor.

Vercel: aplicación y API serverless; región configurable. Documentos grandes se suben directamente a Supabase Storage.

Recomendación: documentar regiones en el registro de tratamiento.

↑ Índice · ↑ Languages

4. Autenticación y sesiones

- Supabase Auth con PKCE y cookies SSR. Producción: **https://fluxehra.ai**.
- Bearer JWT validado en servidor (401 si inválido).
- Sincronización de perfil sin asignar roles en cliente.
- Turnstile opcional anti-bots: login SU global (Setup) y login org por organización (IT Link, `lib/turnstile-settings.ts`).
- Cierre por inactividad configurable (1–60 min).
- Enlaces de invitación/restablecimiento de un solo uso; `/reset-password` con PKCE; onboarding admin org vía `POST /api/admin/send-email`.
- SSO OIDC opcional por organización; redirecciones de error OIDC vía `buildAppRedirectUrl` + `APP_URL`.
- **Solicitud de datos del empleado (nuevas contrataciones):** flujo público con token (`/employee-data-request/{token}`), hash SHA-256 con `EMPLOYEE_DATA_REQUEST_TOKEN_PEPPER`, APIs `/api/public/employee-data-request/*` con rate limit `public_auth`.
- **MFA TOTP (admin / super_admin):** inscripción en `/security/mfa` y **step-up en login** (TOTP tras contraseña) para JWT `aal2`; con `REQUIRE_MFA_FOR_PRIVILEGED_ROLES=true` las APIs exigen `aal2` (403 `MFA_REQUIRED`).
- **Política de contraseña:** `/reset-password` exige ≥ 20 caracteres con mayúscula, minúscula, dígito y carácter especial (`lib/password-policy.ts`) — alinear la configuración Auth de Supabase Dashboard.
- HTML de invitación vía `lib/html-escape.ts` (sin jsdom en Vercel).

↑ Índice · ↑ Languages

5. Roles y control de acceso

Rol	Descripción
Super Admin	Administración global de la plataforma.
Org Admin	Administración de una organización.
Domain Admin	Acceso por módulo vía `role_permissions`.
HR	Ámbito geográfico opcional; autoservicio HR.
Manager	Cadena jerárquica; autoservicio manager.
Empleado	Autoservicio con visibilidad de campos configurable.

APIs servidor y middleware `proxy.ts` aplican el alcance organizacional. **API en denegación por defecto:** todas las rutas `/api/*` requieren sesión salvo lista blanca explícita.

↑ Índice · ↑ Languages

6. Aislamiento multi-inquilino y RLS

RLS en PostgreSQL como frontera principal. Cliente con JWT; service role solo tras autorización en servidor. **Core HR (migración `20260606120000`):** políticas tenant-scoped en `employees`, `employee_historical_data`, `organizations`.

↑ Índice · ↑ Languages

7. Proxy / suplantación

JWT identifica siempre al usuario real. Sesiones proxy en servidor (`proxy_sessions`, 12 h). **Revalidación continua** en cada API protegida (`proxy-api-guard.ts`). Auditoría con registro servidor. No amplía privilegios más allá de RLS.

↑ Índice · ↑ Languages

8. Protección de datos y RGPD

Mecanismos técnicos (visibilidad, retención, cifrado, eliminación en cascada). Cumplimiento legal del responsable del tratamiento. Subencargados: Supabase y Vercel. Aviso de cookies.

↑ Índice · ↑ Languages

9. Visibilidad y retención

Matriz por campo: ninguno, lectura, edición, flujo de trabajo HR. Campos sensibles restringidos. El flag `sensitive: true` activa el **cifrado en base** al guardar (sección 10), no solo en interfaz. Política virtual para datos locales por país. Validación en autoservicio.

↑ Índice · ↑ Languages

10. Cifrado — campos sensibles en base de datos

HTTPS/HSTS en tránsito. Cifrado de infraestructura Supabase en reposo.

Cifrado aplicativo efectivo: los campos con `sensitive: true` no son solo de interfaz — se cifran **antes del INSERT** en PostgreSQL como `ENC2::` (**AES-256-GCM**, IV aleatorio, tag de autenticación, clave derivada por scrypt de `ENCRYPTION_KEY` en servidor). Los valores antiguos `ENC::` siguen siendo legibles (retrocompatibilidad).

- Almacenamiento: columna JSON `employee_historical_data.data`.
- Rutas: contratación, actualización, mass upload — `updateEmployeeData` / APIs `/api/crypto/encrypt`.
- Lectura: descifrado solo vía `/api/crypto/decrypt`; los roles no privilegiados solo pueden descifrar valores de su propio expediente o de su equipo.
- Visibilidad ≠ cifrado: la matriz Data Visibility controla quién ve; `sensitive` controla cómo se persiste.

Límites: solo campos marcados sensibles; documentos binarios usan Storage, no `ENC2::`.

↑ Índice · ↑ Languages

11. Claves API y secretos

Clave anónima (cliente), service role (servidor), ENCRYPTION_KEY, CRON_SECRET, hash de claves de integración, SMTP/Turnstile en configuración. Solo Super Admin modifica ajustes globales.

↑ Índice · ↑ Languages

12. Pista de auditoría

Tabla `audit_logs`, catálogo `AUDIT_EVENTS` (incl. `CORE_HR_HIRE` / `CORE_HR_UPDATE` / `CORE_HR_TERMINATE`), exportación CSV + JSON de evidencia de cumplimiento, vistas de página deduplicadas.

↑ Índice · ↑ Languages

13. Almacenamiento documental

Bucket `documents` **privado** (sin lectura pública). Lecturas vía `/api/documents/file/…` con autorización por visibilidad de campo, árbol jerárquico e inscripciones Learning. **Puerta de enlace Storage** (`lib/storage-ops.ts`) + `isUnsafeStoragePath` (anti-traversal, `%2e`) y `assertStoragePathUnderOrg` . Subidas mediante URLs firmadas tras autorización. Foto (`employee_picture`) visible para miembros de la org (organigrama).

↑ Índice · ↑ Languages

14. Notificaciones e informes de errores

HTML sanitizado, SMTP TLS, cron protegido. Bugs manuales: escalado in-app HR → Org Admin → SU (sin correo). Errores automáticos de cliente: correo SMTP configurable. **ISO A.8.11:** `redactSensitivePii` en canales bug/error; CSV/API/SFTP/DSAR/UI autorizados siguen en claro tras descifrado.

↑ Índice · ↑ Languages

15. Cabeceras HTTP

CSP con nonce por petición (`lib/csp.ts` , aplicada en `proxy.ts`): en producción `script-src` no permite `'unsafe-inline'` ni `'unsafe-eval'` — nonce aleatorio + `'strict-dynamic'` ; `style-src` `'unsafe-inline'` se mantiene como riesgo residual documentado. `X-Powered-By` suprimido. Además HSTS, X-Frame-Options, Permissions-Policy. **ISO A.8.8:** pestaña SU Admin **Components Update** + SLA + CI `npm run audit:deps` (falla en high/critical). Rate limiting global /api durable por usuario (tabla `api_rate_limits` ; 600/3000 archivos/30 public_auth/120/60 req/min). **Vercel Analytics / Speed Insights** (CSP: `vitals.vercel-insights.com`). **ISO A.8.21:** protección SSRF en webhooks, OIDC/SAML, SFTP y **SMTP** (validación al guardar IdP). Reproductor SCORM endurecido. Regiones: Supabase eu-west-1 + Vercel dub1.

↑ Índice · ↑ Languages

16. Lista de verificación IT

1. RLS activo (migración Core HR `20260606120000`).
2. Sin flags TLS inseguros en producción.
3. Regiones eu-west-1 + dub1 documentadas.
4. CRON_SECRET fuerte.
5. Turnstile si aplica.
6. Pruebas IDOR.
7. Monitorizar 429 y 403 MFA_REQUIRED.
8. **MFA:** TOTP en Supabase Dashboard; `REQUIRE_MFA_FOR_PRIVILEGED_ROLES=true` en Vercel; inscripción en `/security/mfa`.
9. Política de contraseña: confirmar ≥ 20 + complejidad en la app y en Supabase Auth Dashboard.

↑ Índice · ↑ Languages

17. Brechas residuales

- Auditoría Core HR hire/update/terminate; purga de retención real; DSAR export/anonimización; MFA por defecto en producción (admin/super_admin/hr); política de contraseña ≥20; Trust Center `/trust`. **Agosto 2026 (rev. 6):** SIEM Axiom (`SECURITY_EVENT`) — BOLA/tenant, fuerza bruta por IP, high-signal (cron/RLS/HMAC/SSRF). **Agosto 2026 (rev. 7):** Anexo A A.8.8 / A.8.11 / A.8.21 Implementado. **Agosto 2026 (rev. 8):** pack SMSI Foil Envie firmado. **Agosto 2026 (rev. 9):** A.5.6 Security Watch + A.8.23 N/A compensado — Anexo A **77 Implemented**. **Agosto 2026 (rev. 10):** HMAC de exportación por clave de integración (`fx_…`); fin del auto-join por dominio e-mail; alcance HR en lectura directa de documentos; grant SU (datos) vs deploy (config) aclarados. Residuales: DPA cliente (A.5.20/A.5.31); revisión independiente (A.5.35); APM Sentry opcional.

Reforzado recientemente (junio–agosto 2026, rev. 10): cabeceras HTTP; API deny-by-default; rate limit; RLS Core HR; MFA TOTP; cifrado AES-256-GCM; bucket documents privado; regiones eu-west-1 + dub1; Employee data request; pack ISO (columna ownership); DSAR y retención; SIEM Axiom; CI vulnerabilidades y SSRF SMTP/IdP; pack SMSI firmado; Security Watch (CERT-FR/CISA/OSV); export HMAC por integración.

contact@fluxehra.ai

↑ Índice · ↑ Languages

Sicherheits- & Compliance-Whitepaper

Multi-Mandanten-HRIS-Plattform — Flex HR

Zielgruppe: IT-Abteilungen, CISO, DSB, Prüfer

Dokumentversion: August 2026 (Rev. 10) — Export-HMAC pro Integrationsschlüssel; kein Domain-Auto-Join; HR-Dokumenten-Scope; SU-Grant vs Deploy klargestellt

---

Inhaltsverzeichnis

1. Einleitung

Dieses Whitepaper beschreibt Sicherheits-, Zugriffs- und Datenschutzmaßnahmen von Flex HR (Multi-Organisations-HRIS) für IT-Teams mit sensiblen HR-Daten.

ISO-Alignment-Pack (Ergänzung, EN/FR): vollständiger Katalog Anhang A ISO/IEC 27001:2022 (93 Kontrollen) — iso-alignment-fluxehra.html — keine ISO-Zertifizierungsbehauptung.

Prinzip: Der Browser ist nicht vertrauenswürdig; Rollen und sensible Operationen werden serverseitig oder per PostgreSQL-RLS unter dem Benutzer-JWT durchgesetzt.

- Supabase Auth (PKCE, JWT).
- Mandantenisolation via `organization_id` und RLS.
- RBAC mit `users` und `role_permissions`.
- Audit in `audit_logs`.
- Feldverschlüsselung (`ENC2::`, AES-256-GCM).

↑ Inhaltsverzeichnis · ↑ Languages

2. Technische Architektur

Frontend: Next.js auf Vercel. **Backend:** Next.js-API-Routen. **Daten:** Supabase (PostgreSQL, Auth, Storage). Shared-Schema-Multi-Tenancy mit RLS. Bitemporale Mitarbeiterhistorie.

↑ Inhaltsverzeichnis · ↑ Languages

3. Hosting & Datenstandort

Supabase: Projektregion bei Erstellung wählbar (z. B. EU: Frankfurt, Paris) — bestimmt primären Speicherort. Anon-Key im Client; `service_role` nur serverseitig.

Vercel: Web-App und serverlose APIs; Region konfigurierbar. Große Dokumente direkt zu Supabase Storage.

Empfehlung: Regionen im Verarbeitungsverzeichnis dokumentieren.

↑ Inhaltsverzeichnis · ↑ Languages

4. Authentifizierung & Sitzungen

- Supabase Auth mit PKCE und SSR-Cookies. Produktion: **https://fluxehra.ai**.
- Bearer-JWT serverseitig validiert (401 bei Ungültigkeit).
- Profil-Sync ohne Rollenzuweisung im Client.
- Optionales Turnstile: global SU-Login (Setup) und pro Organisation (IT Link, `lib/turnstile-settings.ts`).
- Inaktivitäts-Logout (1–60 Min.) konfigurierbar.
- Einmalige Einladungs-/Reset-Links; `/reset-password` mit PKCE; Org-Admin-Onboarding via `POST /api/admin/send-email`.
- Optionales OIDC-SSO pro Organisation; OIDC-Fehler-Redirects über `buildAppRedirectUrl` + `APP_URL`.
- **Employee data request (Neueinstellungen):** öffentlicher Token-Flow (`/employee-data-request/{token}`), SHA-256-Hash mit `EMPLOYEE_DATA_REQUEST_TOKEN_PEPPER`, APIs unter `/api/public/employee-data-request/*` (Rate-Limit `public_auth`).
- **MFA TOTP (admin / super_admin):** Registrierung unter `/security/mfa` und **Step-up beim Login** (TOTP nach Passwort) für JWT `aal2`; mit `REQUIRE_MFA_FOR_PRIVILEGED_ROLES=true` erfordern APIs `aal2` (403 `MFA_REQUIRED`).
- **Passwortrichtlinie:** `/reset-password` verlangt ≥ 20 Zeichen mit Groß-/Kleinbuchstaben, Ziffer und Sonderzeichen (`lib/password-policy.ts`) — Auth-Einstellungen im Supabase Dashboard angleichen.
- Einladungs-HTML über `lib/html-escape.ts` (ohne jsdom auf Vercel).

↑ Inhaltsverzeichnis · ↑ Languages

5. Rollen & Zugriffskontrolle

Rolle	Beschreibung
Super Admin	Globale Plattformverwaltung.
Org Admin	Verwaltung einer Organisation.
Domain Admin	Modulzugriff über `role_permissions`.
HR	Geografischer Scope; HR Self Service.
Manager	Hierarchiekette; Manager Self Service.
Mitarbeiter	Self Service mit Feldsichtbarkeit.

Server-APIs und Middleware `proxy.ts` erzwingen Organisationsumfang. **Default-Deny-API:** alle `/api/*`-Routen erfordern Session außer expliziter Allowlist.

↑ Inhaltsverzeichnis · ↑ Languages

6. Mandantenisolation & RLS

PostgreSQL-RLS als Hauptgrenze. Client mit JWT; Service Role nur nach Autorisierung. **Core HR (Migration `20260606120000`):** mandantenspezifische Policies auf `employees`, `employee_historical_data`, `organizations`.

↑ Inhaltsverzeichnis · ↑ Languages

7. Proxy / Identitätswechsel

JWT identifiziert immer den echten Benutzer. `proxy_sessions` serverseitig (12 h TTL). **Kontinuierliche Revalidierung** bei jedem geschützten API-Aufruf (`proxy-api-guard.ts`). Audit über Serverdatensatz. Keine Rechte über RLS hinaus.

↑ Inhaltsverzeichnis · ↑ Languages

8. Datenschutz & DSGVO

Technische Mechanismen (Sichtbarkeit, Aufbewahrung, Verschlüsselung, Löschkaskade). Rechtliche Compliance beim Verantwortlichen. Auftragsverarbeiter: Supabase und Vercel. Cookie-Hinweis.

↑ Inhaltsverzeichnis · ↑ Languages

9. Feldsichtbarkeit & Aufbewahrung

Matrix pro Feld: none, view, edit, workflow. Sensible Felder eingeschränkt. `sensitive: true` löst **DB-Verschlüsselung** beim Speichern aus (Abschnitt 10), nicht nur UI. Virtuelle Local-Data-Policy. Self-Service-Validierung serverseitig.

↑ Inhaltsverzeichnis · ↑ Languages

10. Verschlüsselung — sensible Felder in der Datenbank

HTTPS/HSTS in Transit. Supabase-Infrastrukturverschlüsselung at rest.

Wirksame Anwendungsverschlüsselung: Felder mit `sensitive: true` werden **vor dem PostgreSQL-INSERT** als `ENC2::` (**AES-256-GCM**, zufälliger IV, Auth-Tag, scrypt-abgeleiteter Schlüssel aus `ENCRYPTION_KEY`) gespeichert — nicht nur UI-Maskierung. Alte `ENC::`-Werte bleiben rückwärtskompatibel lesbar.

- Speicher: JSON-Spalte `employee_historical_data.data`.
- Schreibpfade: Einstellung, Updates, Mass Upload — `updateEmployeeData` / `/api/crypto/encrypt`.
- Lesen: Entschlüsselung nur über `/api/crypto/decrypt`; nicht privilegierte Rollen können nur Werte aus der eigenen Akte oder dem eigenen Team entschlüsseln.
- Sichtbarkeit vs. Verschlüsselung: Data Visibility = wer sieht; `sensitive` = wie gespeichert.

Grenzen: nur markierte Felder; Dokumente im Storage separat.

↑ Inhaltsverzeichnis · ↑ Languages

11. API-Schlüssel & Geheimnisse

Anon-Key (Client), Service Role (Server), ENCRYPTION_KEY, CRON_SECRET, Integrations-Key-Hashing, SMTP/Turnstile in Einstellungen. Globale Settings nur Super Admin.

↑ Inhaltsverzeichnis · ↑ Languages

12. Audit-Trail

Tabelle `audit_logs`, Katalog `AUDIT_EVENTS` (inkl. `CORE_HR_HIRE` / `CORE_HR_UPDATE` / `CORE_HR_TERMINATE`), CSV-Export + Compliance-JSON, deduplizierte Seitenaufrufe.

↑ Inhaltsverzeichnis · ↑ Languages

13. Dokumentenspeicher

Privater Supabase-Bucket `documents`. Lesezugriffe über `/api/documents/file/…` mit Feldsichtbarkeit und Learning-Enrollments. **Storage-Gateway** (`lib/storage-ops.ts`) + `isUnsafeStoragePath` und `assertStoragePathUnderOrg`. Uploads über signierte URLs. `employee_picture` für Org-Mitglieder sichtbar.

↑ Inhaltsverzeichnis · ↑ Languages

14. Benachrichtigungen & Fehlermeldungen

Sanitisiertes HTML, TLS-SMTP, geschützter Cron. Manuelle Bugs: In-App-Eskalation HR → Org Admin → SU (kein E-Mail). Automatische Client-Fehler: SMTP. **ISO A.8.11:** `redactSensitivePii` auf Bug-/Fehlerkanälen; autorisierte CSV/API/SFTP/DSAR/UI bleiben nach Entschlüsselung Klartext.

↑ Inhaltsverzeichnis · ↑ Languages

15. HTTP-Sicherheitsheader

Nonce-basierte CSP pro Request (`lib/csp.ts`, angewendet in `proxy.ts`): in Produktion erlaubt `script-src` weder `'unsafe-inline'` noch `'unsafe-eval'`. `X-Powered-By` unterdrückt. Dazu HSTS, X-Frame-Options, Permissions-Policy. **ISO A.8.8:** SU-Admin **Components Update** + SLA + CI `npm run audit:deps` (Fail bei high/critical). Globales /api-Rate-Limiting (600/3000 Dateien/30 public_auth/120/60 req/min). **Vercel Analytics / Speed Insights** (CSP: `vitals.vercel-insights.com`). **ISO A.8.21:** SSRF-Schutz für Webhooks, OIDC/SAML, SFTP und **SMTP** (Validierung beim IdP-Speichern). SCORM-Härtung. Regionen: eu-west-1 + dub1.

↑ Inhaltsverzeichnis · ↑ Languages

16. IT-Prüfliste

1. RLS aktiv (Core-HR-Migration `20260606120000`).
2. Keine unsicheren TLS-Flags in Produktion.
3. Regionen eu-west-1 + dub1 dokumentiert; Produktion `https://fluxehra.ai`.
4. Starkes CRON_SECRET.
5. Turnstile global (SU) und/oder pro Org (IT Link) bei Bedarf.
6. IDOR-Tests.
7. 429- und 403 MFA_REQUIRED-Monitoring.
8. **MFA:** TOTP in Supabase Dashboard; `REQUIRE_MFA_FOR_PRIVILEGED_ROLES=true` auf Vercel; Registrierung unter `/security/mfa` und Step-up beim Login für `aal2`.
9. Passwortrichtlinie: ≥ 20 + Komplexität in der App und im Supabase Auth Dashboard bestätigen.
10. Supabase-Redirect-URLs mit `https://fluxehra.ai/**`.

↑ Inhaltsverzeichnis · ↑ Languages

17. Verbleibende Lücken

- Core-HR-Audit für Hire/Update/Terminate; echte Retention-Purge; DSAR Export/Anonymisierung; MFA standardmäßig ON in Produktion (admin/super_admin/hr); Passwortrichtlinie ≥20; Trust Center `/trust`. **August 2026 (Rev. 6):** Axiom-SIEM (`SECURITY_EVENT`) — BOLA/Tenant, Brute-Force pro IP, High-Signal (Cron/RLS/HMAC/SSRF). **August 2026 (Rev. 7):** Annex A A.8.8 / A.8.11 / A.8.21 Implemented. **August 2026 (Rev. 8):** Foil-Envie-ISMS-Pack signiert. **August 2026 (Rev. 9):** A.5.6 Security Watch + A.8.23 N/A kompensiert — Anhang A **77 Implemented**. **August 2026 (Rev. 10):** Export-HMAC pro Integrationsschlüssel (`fx_…`); kein Domain-Auto-Join; HR-Scope bei direkter Dokumentenlesung; SU-Grant (Daten) vs Deploy (Config) klargestellt. Restlücken: Kunden-DPA (A.5.20/A.5.31); unabhängige Prüfung (A.5.35); optionales Sentry-APM.

Kürzlich verstärkt (Juni–August 2026, Rev. 10): HTTP-Sicherheitsheader; Default-Deny-API; Rate-Limiting; Core-HR-RLS; MFA TOTP; AES-256-GCM; privater Documents-Bucket; Region eu-west-1 + dub1; Employee data request; ISO-Pack (Ownership-Spalte); DSAR und Retention; Axiom-SIEM; CI-Vulnerabilities und SMTP/IdP-SSRF; signiertes ISMS-Pack; Security Watch (CERT-FR/CISA/OSV); Export-HMAC pro Integration.

contact@fluxehra.ai

↑ Inhaltsverzeichnis · ↑ Languages