redsmart RH



SECURITY & DATA PROTECTION

Data isolation between tenants & security measures

How RedSmart RH ensures that each organization's information stays separated, protected and under its control.

Technical document · July 2026 · v1.0

RedSmart RH — Labor-compliance platform · by Go4IT

SECTION 00

Executive summary

RedSmart RH is a white-label **multi-tenant** (multi-company) platform: several organizations —called tenants— run on the same infrastructure, yet their data stays strictly separated. This document describes, with evidence verified in production, how we guarantee that isolation and what additional measures protect the information — in particular employees' **sensitive** personal data (health, biometrics, payroll), regulated in Mexico by the LFPDPPP.

CORE IDEA

No request can ask for another tenant's data. The organization's identity never travels as a parameter the client can modify: it is derived from the authenticated credential and enforced across two independent layers — the application and the database engine.

SECTION 01

Multi-tenant model: the concept of "origin"

Every record in the platform carries a tenant tag, the **origin**, that identifies the organization that owns the data. On authentication, the system determines the user's **origin** from their credential — **not** from what the client sends — and propagates it across the whole request. From there, every read and write is scoped to that origin. It is separation **by design**, not a flag that can be requested some other way.

SECTION 02

Two-layer isolation (defense in depth)

Access to any data must pass **two independent controls**. If one failed because of a programming error, the other still stands and no data is exposed.

LAYER 1 · Application

Every query explicitly filters by the **requester's origin** (@Origen), which always comes from the credential. There is no "tenant" parameter the client controls.

▼ and on top of that ▼

LAYER 2 · Database engine (Row-Level Security)

The database itself re-filters every row by the session's active tenant. Even if a query had a bug and "forgot" to filter, the engine **will not return or allow inserting** rows from another tenant. Current coverage: **44 of 44** HR tables.

WHY IT MATTERS

Data is only released if it passes BOTH layers. It is the difference between "we trust the code filters correctly" and "the engine prevents it even if the code fails".

SECTION 03

How the tenant identity flows

On every request, the organization's identity is set server-side and travels down to the data engine:

1. **Incoming request**
   Arrives from the web app, the REST API or an AI assistant (MCP).
2. **Authentication**
   The credential is validated: SSO session (JWT), API key or OAuth token.
3. **Tenant is extracted**
   The **origin** is obtained from the credential — never from the content the client sends.
4. **Secure context is set**
   The server sets the active tenant on the database connection (SESSION_CONTEXT).
5. **Layer 1 — application filter**
   The query asks only for the requester's **origin**.
6. **Layer 2 — engine RLS**
   The database re-filters and blocks any foreign row.
7. **Result**
   Only the owning organization's data is returned.

SECTION 04

Access surfaces and how they are isolated

All access paths share the **same** tenant mechanism described above. Third-party and AI integrations are **read-only**.

Surface	Authentication	How it is isolated
Web app employee / HR	SSO session (JWT OpenIddict)	Identity → origin → the 2 layers
REST API /api/ext	API key (rsk_..., SHA-256 hash)	Same mechanism · read-only
MCP server AI assistants	API key or OAuth 2.1 (PKCE)	Same mechanism · read-only
Outbound webhooks	HMAC-SHA256 signature per tenant	Each event is delivered only to its own tenant's webhook

SECTION 05

Empirical verification (not just theory)

We tested isolation directly against the production database, forcing the tenant context in the engine and counting visible rows across the HR tables (courses, announcements, NOM-035, internal work rules, breaks, API keys, etc.):

Scenario	What it represents	Result
Context = foreign tenant	A third party trying to read data that isn't theirs	**0 rows across ALL tables — blocked**
Context = real tenant	The organization that owns the data	Sees only **its own** data
Internal processes	System tasks (sends, maintenance) with no user session	Run with controlled access (safe fail-open); they do not expose data to clients

CONCLUSION

A tenant that is not the owner sees zero. Isolation does not depend on "the code behaving well": the database engine enforces it.

SECTION 06

Additional security measures

IDENTITY & ACCESS

Our own single sign-on (SSO OpenIddict) with passwords protected by BCrypt, **passkeys** (WebAuthn, passwordless), lockout on failed attempts, session revocation within minutes, and sliding sessions.

TRANSPORT & SURFACE

Forced HTTPS with HSTS, OWASP security headers + CSP, server technology hidden, and **rate limiting** per IP and per API key, with anti-enumeration protection on the time-clock kiosk.

DATA AT REST & SENSITIVE DATA

Encryption at rest (Azure SQL TDE); **field-level encryption** (Always Encrypted) for the most sensitive record data; and an **access log** for sensitive data (who viewed or changed what).

CONTROLLED INTEGRATIONS

API keys stored as a hash and instantly revocable (read-only); OAuth 2.1 dedicated to the AI assistant — with PKCE, read-only and **isolated** from the platform SSO —; and **signed** webhooks (HMAC) with retries.

FILES & MEDIA

Private Azure Blob storage accessed via short-lived temporary links (SAS) and a managed identity; no permanent public URLs.

BIOMETRIC DATA (time-clock face verification)

Express, revocable consent under the LFPDPPP, and **automatic purge** of images outside the retention window.

SECTION 07

Alignment with the LFPDPPP

HR data includes **sensitive** categories (health and leaves, biometrics, affiliation). The platform was designed in alignment with Mexico's Federal Law on Protection of Personal Data Held by Private Parties: express consent for sensitive data, minimization and separation per organization, access logging, encryption of critical fields, and biometric retention/purge.

Honesty note: the official determination of obligations —for example the IMSS occupational-risk premium— rests with the authority. RedSmart **records and estimates**; it does not replace IMSS/SUA.

SECTION 08

Glossary of terms

Term	What it means
Tenant	Each client organization on the platform; its data is separated from the others.
Origin	Internal tag marking which tenant each piece of data belongs to.
RLS	Row-Level Security: per-row filtering enforced by the database engine itself.
SSO	Single sign-on; RedSmart's central identity system.
MCP	Protocol that lets AI assistants query data in a standardized way.
API key	Secret key (rsk_...) that authenticates a third-party system; read-only and revocable.
OAuth 2.1 · PKCE	Secure standard for apps like ChatGPT or Claude to connect without sharing passwords.
HMAC	Cryptographic signature guaranteeing a webhook comes from RedSmart and was not tampered with.
SAS	Temporary link to a stored file; expires on its own.
TDE · Always Encrypted	Encryption at rest (whole disk) and field-level encryption (specific data).
LFPDPPP	Mexico's Federal Law on Protection of Personal Data Held by Private Parties.

In summary

Separation by design + two independent, **verified** layers + hardened credentials and transport + sensitive data encrypted and audited. RedSmart RH treats your people's information as the most critical thing — because it is.

Technical questions or a security review?

Go4IT · RedSmart RH — we are glad to expand on any point of this document with your IT or compliance team.