COMPANY
REFI2 Labs Inc.

OFFICE
100 King St W, Suite 6200
1 First Canadian Place
Toronto, Ontario M5X 1B8, Canada

CONTACT
+1 (905) 536-6968
info@refi2.com
refi2.com

REFI2 Security Architecture

Overview

REFI2 data is encrypted in transit and at rest on every layer, access is gated by strong identity, and signed documents are cryptographically sealed. The controls below are built against the AICPA SOC 2 Trust Services Criteria for Security, Availability, and Confidentiality. No independent SOC 2 examination has been performed yet. This document describes the controls an auditor would test.

The platform has three applications. The investor portal serves the web application, the on-chain dashboard, and the administrative console. The data room shares confidential documents with investors. The e-signature service executes agreements. All three run on managed cloud infrastructure whose providers hold their own SOC 2 Type II and ISO/IEC 27001 attestations, listed in the Infrastructure section.

Encryption in transit

Every connection to the platform is encrypted with TLS, and all three applications negotiate TLS 1.3 with modern clients (verified October 14, 2026).

PROPERTY	VALUE
Protocol negotiated	TLS 1.3; anything older than TLS 1.2 is refused on all three applications
Cipher suites	TLS_AES_256_GCM_SHA384, TLS_AES_128_GCM_SHA256 (AEAD, forward secret)
Key exchange	Ephemeral ECDHE, so a future key compromise cannot decrypt recorded traffic
Plain HTTP	Permanently redirected to HTTPS (308)
HSTS	`max-age=63072000` (two years) on the portal and the data room, one year with subdomains on the signing service
Certificates	Issued and rotated automatically by the edge network, never handled by staff

TLS terminates at a global edge network that also absorbs volumetric DDoS traffic before it reaches the application. Traffic between the application runtime and the managed databases travels either over TLS or over the hosting provider’s encrypted private network, and never crosses the public internet in clear text.

TLS 1.3 removes every cipher and handshake mode that earlier versions were attacked through (RC4, CBC padding oracles, RSA key transport, and renegotiation). With AEAD ciphers, any modification of data in flight is detected and the connection is dropped.

Encryption at rest

The storage layer encrypts all stored data. Secrets that must stay protected even if the whole database leaks are encrypted a second time by the application.

DATA	WHERE IT LIVES	PROTECTION
Data room documents	Private object storage, no public listing	AES-256-GCM at rest; served only through signed URLs that expire after one hour
Identity verification files	Private blob storage	AES-256 at rest; never exposed through a public URL
Platform secrets and API keys	Encrypted environment store of the hosting platform	AES-256 at rest; injected at runtime, never committed to source control
Two-factor secrets and backup codes	Relational database, application-encrypted column	XChaCha20-Poly1305 (256-bit key, 192-bit random nonce)
Document access passwords	Relational database, application-encrypted column	AES-256-GCM with a per-record random IV; any tampering fails decryption
User passwords	Relational database	bcrypt, one-way, never stored or logged in plain text
Relational data (PostgreSQL, libSQL)	Managed database services	Provider-managed storage encryption under their SOC 2 Type II controls

Application-layer encryption keys live only in the runtime environment, separate from the database. A copy of the database alone is not enough to recover a second-factor secret or a protected link password.

XChaCha20-Poly1305 is an authenticated cipher, so any change to the ciphertext makes decryption fail instead of returning altered data. Its 192-bit nonce is generated randomly per value, which removes the nonce-reuse risk that affects shorter-nonce ciphers.

The identity verification flow keeps personal data on the server. The administrative console receives masked values (“M. W.”, partially hidden emails), and revealing one full record is a separate, logged request.

Identity and access control

Every request is authenticated before it reaches application logic, and the most powerful role on the platform cannot be granted by a single person.

AREA	AUTHENTICATION	AUTHORIZATION
Investor portal	Managed identity provider (OAuth 2.0/OIDC, short-lived session tokens verified server side, breached-password screening, mandatory multi-factor authentication) or a Solana wallet signature	A global gate denies every route by default; the few public routes are listed explicitly in code and authenticate inside the handler
Administrative console	Same identity layer	Membership in an on-chain multisignature wallet; the server resolves the member list itself and ignores anything the browser claims
Data room, staff side	Google OAuth	Login restricted to an allowlist of corporate domains and addresses
Data room, viewer side	Email-gated links, optional one-time email verification	Per-link expiry, download disabled by default, optional NDA click-through and watermark with the viewer's identity
E-signature service	Password (bcrypt), Google or Microsoft OAuth, OIDC, WebAuthn passkeys, and TOTP two-factor	Per-document recipient tokens; each signer sees only the envelope addressed to them
Programmatic access	Bearer personal access tokens with a distinct prefix, or OAuth tokens	Scoped to the issuing user and checked on every call

Administrative authority over the on-chain protocol (mint, freeze, and emergency pause) sits in an M-of-N multisignature wallet. No single key, laptop, or employee can execute a privileged action. The same member list decides who may open the administrative console, so revoking a member on chain removes their console access as well.

Passkeys (WebAuthn/FIDO2) bind the credential to the domain, which makes phishing pages unable to replay it. Login and sensitive endpoints are rate limited, and forms open to the public carry bot protection with a privacy-preserving challenge.

Access to production systems, the administrative console, and the multisignature wallets is reviewed manually every quarter. With a small team every grant is checked by name, and access is revoked immediately when someone leaves. Enterprise single sign-on (SAML) and SCIM provisioning are not offered, because the number of accounts on each side is small enough to manage directly.

Document integrity

A signed agreement on the platform cannot be altered, backdated, or have an event silently removed without the change being detectable.

Each completed document is sealed with an X.509 digital signature over the PDF. Any byte changed after sealing breaks the signature, and standard PDF readers show the document as modified. Every seal carries an RFC 3161 trusted timestamp from an independent timestamp authority, proving when the

document existed in its final form. In production the service refuses to seal a document if no timestamp authority is reachable.

The signing audit trail is tamper-evident. Every event (sent, opened, signed, declined, or completed) is written with the actor's identity, time, IP address, and user agent, and hashed with HMAC-SHA256 together with the hash of the previous event. Each document keeps an anchor over the event count and the latest hash.

ATTEMPTED TAMPERING	RESULT
Editing an event's content	Its hash no longer matches
Backdating an event	The timestamp is inside the hash, so the chain breaks
Deleting an event	The next event's link to its predecessor breaks, and the anchor count no longer matches
Reordering events	Each event commits to the one before it
Rewriting the whole chain	Requires the HMAC key, which is held outside the database

The chain logic has a dedicated test suite that runs in CI against a disposable database on every change, guarded so it can never run against a production database.

Application security

The applications are hardened against the OWASP Top 10 classes of attack with layered browser and server controls.

CONTROL	IMPLEMENTATION
Content Security Policy	Enforced on the data room and the e-signature service. The e-signature service uses a per-request nonce with `strict-dynamic`, `object-src 'none'`, `base-uri 'self'`, `form-action 'self'`; CI verifies the allowed inline style hashes on every change
Clickjacking	`X-Frame-Options: SAMEORIGIN` and `frame-ancestors` restrictions on every application, DENY on the investor portal; only purpose-built embed routes may be framed
MIME sniffing and referrer leakage	`X-Content-Type-Options: nosniff`, restrictive `Referrer-Policy`
CSP monitoring	Violations are reported to a dedicated endpoint for review
Search exposure	Document viewer routes send `X-Robots-Tag: noindex`
Rate limiting	Middleware on authentication and sensitive endpoints, with scheduled cleanup
Bot protection	Privacy-preserving challenge plus honeypot fields on public forms
Input validation	Typed schemas (Zod) validate API payloads before they reach business logic
Data access	ORM-parameterized queries (Prisma, Kysely), which rules out SQL injection through string building
Sensitive data in the browser	Personal data is masked on the server; the browser never receives a full value it was not explicitly authorized to see

Browser features the platform does not use, such as the camera and geolocation, are disabled through Permissions-Policy. Secrets never reach the client bundle. Server-only modules are marked so the build fails if one is imported into browser code.

Change management and secure development

Code reaches production only through version control and automated checks, and nobody edits a running server by hand.

1. Work happens on feature branches and merges into the main branch through pull requests with a standard template.
2. Pre-commit hooks lint staged files and enforce structured commit messages.
3. Continuous integration runs on every pull request and every push to main and runs type checking, linting, and unit and integration tests. The on-chain programs additionally run Clippy with warnings treated as errors and `cargo deny` against known-vulnerable and disallowed dependencies.
4. Toolchain versions are pinned (Rust, Solana, Anchor, Node, and the CI runner image), and the build fails if an installed version drifts.
5. Deployments are built from the main branch by the hosting platform, producing immutable versions that can be rolled back instantly.

CI is built on least privilege. Workflow tokens are read-only, every secret is scoped to the single step that needs it, and repository credentials are not persisted for later steps. Secrets are excluded from source control in every repository, including signing certificates and private keys.

Components built on open source projects publish their modified source as their licenses require, so the code that handles documents is open to independent inspection.

Infrastructure and subprocessors

The platform runs no self-managed servers. Compute, storage, databases, and identity are delegated to providers that are independently audited, so physical security, hardware disposal, host patching, and network isolation are covered by their attestations (the standard carve-out model in a SOC 2 report).

Applications run as serverless functions and isolated containers built from version-controlled images.

LAYER	PROVIDER ATTESTATIONS	REPORT ACCESS
Serverless hosting and edge network	SOC 2 Type II, ISO/IEC 27001, PCI DSS	Vercel security
Container hosting and managed PostgreSQL	SOC 2 Type II, SOC 3, storage-level encryption at rest	Railway trust center, compliance
DNS, DDoS protection, object storage	SOC 2 Type II, ISO/IEC 27001	Cloudflare compliance, object storage encryption
Identity and session management	SOC 2 Type II, HIPAA	Clerk trust center, security
Edge database (libSQL)	SOC 2 Type II, volume-level encryption at rest	Turso trust center
OAuth and key management	SOC 2 Type II, ISO/IEC 27001	Google Cloud compliance
Offsite encrypted backups	SOC 2 Type II, ISO/IEC 27001	AWS compliance

Full provider SOC 2 reports are distributed under NDA and can be requested directly through each trust center.

Backups and disaster recovery

Production databases are dumped daily, encrypted before they leave the hosting environment, and stored in AWS S3, separate from the primary infrastructure.

PROPERTY	VALUE
Backup frequency	Daily
Encryption	Encrypted before upload with a key held outside the backup store; AES-256 at rest in S3
Location	AWS S3, independent of the primary hosting providers
Retention	30 days
Recovery point objective (RPO)	24 hours
Recovery time objective (RTO)	4 hours

Applications are stateless and redeploy from version control in minutes, so recovery time is driven by the database restore. A provider outage is handled by restoring the latest dump into a fresh managed database and pointing the application at it.

Logging, monitoring and incident response

Application and platform logs are centralized and kept for one year. Alerts on errors and security-relevant events reach the engineering team directly.

Incidents follow a written response plan with severity levels, a named incident lead, containment, root-cause analysis, and a written post-incident review. Customers affected by a confirmed security incident involving their data are notified within 72 hours.

Data retention and deletion

Data is kept only as long as the service or the law requires.

DATA	RETENTION
Identity verification (KYC) records	5 years after the relationship ends, as anti-money-laundering rules require
Signed agreements and their audit trail	5 years after the relationship ends
Other personal data	Deleted within 30 days of a verified deletion request, unless the law requires keeping it
Application and platform logs	1 year
Database backups	30 days, so deleted data leaves the backups within that window

Vulnerability disclosure

Security issues can be reported privately to security@refi2.com. Every application publishes an RFC 9116 `security.txt` at `/.well-known/security.txt` with the same contact.

SOC 2 criteria mapping

The technical controls above map to the following 2017 Trust Services Criteria (revised points of focus, 2022).

CRITERION	REQUIREMENT	PLATFORM CONTROL
CC6.1	Logical access security, encryption of data at rest	Default-deny authentication gate, multisig-rooted admin access, AES-256 and XChaCha20-Poly1305 encryption
CC6.2, CC6.3	Provisioning, modification and removal of access	Allowlisted staff login, on-chain membership as the source of truth for admin roles, quarterly manual access reviews
CC6.6	Protection against threats from outside the boundary	Edge DDoS protection, rate limiting, bot challenge, CSP and framing controls
CC6.7	Restricting and protecting data in transmission	TLS 1.3, HSTS, signed short-lived download URLs
CC6.8	Preventing unauthorized or malicious software	Pinned toolchains, dependency policy checks, CI on every change
CC7.2	Monitoring for anomalies	Tamper-evident audit chain, CSP violation reporting, logged reveals of personal data
CC8.1	Authorized, tested and approved changes	Pull request workflow, automated checks, immutable deployments with rollback
CC9.2	Vendor risk management	Subprocessors limited to providers with SOC 2 Type II or ISO/IEC 27001 attestations
C1.1	Identifying and protecting confidential information	Private storage, expiring links, NDA gating, watermarks, server-side masking
PI1 (supporting)	Integrity of processed records	Digitally sealed PDFs with RFC 3161 timestamps
CC2.3	Communication with external parties	Published security contact and security.txt on every application
CC7.3 to CC7.5	Evaluating, responding to and recovering from incidents	Centralized logs, alerting, written incident response plan, 72-hour customer notification
A1.2, A1.3	Backup, recovery and recovery testing	Daily encrypted offsite backups, RPO 24 hours, RTO 4 hours
C1.2	Disposal of confidential information	Retention schedule, deletion within 30 days of request, 30-day backup expiry