Clan information security policy

Version 1.0 · Effective 2026-09-21 · Owner: Louis Anderson (founder; responsible for information security) · louisanderson2301@gmail.com

This policy is short because the organisation is one person and a system that enforces most of its own rules. It says who is responsible, what the rules are, and where each rule is enforced in code or configuration, so that a reader can check the claim rather than take it on trust. It is reviewed at least annually and whenever the architecture changes; the review is recorded in the changelog at the end.

1. Scope and responsibility

Clan is a family-finance application (`github.com/L2301/clan`) serving `https://clan-household-2026.web.app`. It holds financial information members enter or connect, including bank account data obtained through Plaid.

The founder is the security owner: accountable for this policy, for responding to incidents, and for every item in the SLAs below. There are no other staff or contractors. Where a control is performed by a person, it is performed by the founder; where it is performed by the system, the section names the file.

2. Data classification

Class	Examples	Handling
Credentials	Plaid access tokens, provider API keys, database passwords, TLS keys, TOTP secrets	Never in Git, logs, chat, or the browser. Encrypted at rest with AES-256-GCM under a key held only in the app's environment (`src/secrets/vault.ts`, `src/server/mfa.ts`). Readable only by the dedicated `clan_vault` database role.
Financial data	Balances, transactions, commitments, conversations, documents	Stored under row-level security keyed to the owning member; encrypted at rest at the disk level; retained per §7; deleted on request per §8.
Identity	Email, Google subject, session hashes	Held in tables only the `clan_identity` role can read; sessions stored as hashes only.
Operational	Request logs, error logs, audit log	No financial values or message text in logs. 30-day retention.

3. Access control

Policy. Access to production assets and data is granted to named identities only, at the minimum privilege needed, and is reviewed whenever it changes.

- **People.** One person holds production access, through a personal Google account with a phishing-resistant second factor (passkey or hardware key) required. This account governs Google Cloud (`monasterion-network`, `clan-household-2026`), Firebase, the GCE host, GitHub, the Plaid dashboard and OpenRouter. A second person added in future receives access through Google Cloud IAM roles, never shared credentials, and is removed from IAM the day their involvement ends.
- **Host.** SSH to `eco-node-1` is key-only: `PasswordAuthentication no`, `PermitRootLogin no`, `KbdInteractiveAuthentication no` (verified with `sshd -T`). Keys are held in the founder's macOS keychain.
- **Services (non-human).** Cloud Run → application traffic is authenticated by a random gateway key and a pinned TLS certificate (`deploy/gateway`, `deploy/shared/install.sh`); requests without the key are refused before reaching the application. Cloud Build runs as a dedicated `clan-build` identity with write access only to Clan's Artifact Registry repository. Google sign-in uses OAuth with PKCE, single-use state, nonce and a browser-bound cookie (`src/server/google.ts`).
- **Database roles.** Three roles, each fenced by grants and row-level security: `clan_app` (financial data, no identity or vault tables), `clan_identity` (sessions, credentials, second-factor secrets, no financial tables), `clan_vault` (encrypted credentials only). Migrations 020, 024, 025,
  50. Tests pin the separation (`test/identity.test.ts`, `test/tenancy.test.ts`).
- **Application roles.** Household creator, funder, recipient, observer. Visibility is a property of each record (`isParty` / `isSelf`, `src/commitments/view.ts`), enforced under RLS, so no role can read another member's private data.
- **Consumer second factor.** Members may enrol a TOTP authenticator (RFC 6238) with single-use recovery codes. Once enrolled it is required at every sign-in, and the whole API is closed to an unverified session (`src/server/api.ts`). Enrolment is **required before any bank connection can be started** (`src/plaid/tools.ts` → `linkEligibility`).

4. Encryption

- **In transit.** TLS 1.2 minimum everywhere: Firebase Hosting refuses TLS 1.0 and 1.1 and sends HSTS with preload; the Cloud Run gateway connects to the host over TLS with a pinned CA;

the host relay is configured `minVersion: 'TLSv1.2'` (`deploy/gateway/tls-relay.mjs`). Outbound calls to Plaid, Google, OpenRouter and Anthropic are HTTPS only.
- **At rest.** All persistent disks on the GCE host are encrypted by Google Cloud by default (Google-managed keys); this covers the database, uploaded documents, browser state and backups. Credentials are additionally encrypted at the application layer as described in §2.
- **Keys.** The vault key, database passwords, gateway key and TLS material live in `/opt/clan/private` (mode 700, root-only) and are never copied into Git, images, logs or chat.

5. Secure development and vulnerability management

SLA. A vulnerability that affects Clan — in a dependency, a base image, the host operating system, or Clan's own code — is patched or mitigated **within 2 days** of its becoming known, at any severity, and the fix is deployed to production within the same window. If a fix is unavailable, the affected component is disabled or isolated within the same 2 days.

How that is met:

- **Dependencies.** Dependabot is configured for npm, Docker base images and GitHub Actions with a daily schedule (`.github/dependabot.yml`). Every push and every day, CI runs `npm audit` and fails on any finding of moderate severity or above (`.github/workflows/security.yml`). Production dependencies are audited clean at each release.
- **Host.** `unattended-upgrades` is enabled on `eco-node-1` and applies Debian security updates daily (verified active). The host runs a supported Debian release; end-of-life dates for Debian, Node.js and the Playwright base image are tracked, and no component is run past its end of life.
- **Images.** Containers are rebuilt from current base images on every publish (`deploy/cloudbuild.yaml`); a publish is made at least monthly even with no code change, so base-image fixes reach production.
- **Code.** Every change goes through the test suite (300+ tests) including tests that pin the disclosure boundary, session controls, CSRF, RLS role separation, and the second factor. Financial invariants are enforced by database constraints and triggers, not by application code.
- **Endpoint devices.** The founder's Mac runs macOS with automatic updates enabled, FileVault on, and no shared accounts. There are no other endpoints.

6. Logging, monitoring and incident response

- The application logs requests and errors without financial values or message text; an audit log records security-relevant actions (sign-in events, invitations, message supersession, erasure).
- **An incident** is any suspected unauthorised access to, or loss of, credentials or member data. On discovery the founder will, within 24 hours: revoke affected credentials (Plaid tokens via `/item/remove`, sessions via `logout-all`, API keys at the provider); contain the cause; and preserve logs. Affected members are notified within 72 hours of confirmation, with what happened, what data was involved, and what has been done. Plaid is notified as their agreement requires. A written post-mortem is added to `docs/incidents/`.

7. Data retention

Data	Retained	Then
Members' financial records and conversations	While the account exists	Deleted on erasure (§8)
Sessions, OAuth state, sign-in attempts, delivered chat jobs	Until expired / complete	Purged within 30 days (`scripts/retention.js`, run daily)
Request and error logs	30 days	Rotated by Docker's log driver
Backups	≤ 30 days	Deleted; the host holds no backup older than 30 days
Audit log	2 years	Contains no financial values or message text

8. Deletion

Any member can delete their account from **Settings → Account → Delete my account**, confirmed with their second factor. Deletion is immediate and implemented in `src/identity/erase.ts`:

1. Plaid items are removed at Plaid (`/item/remove`) before anything local is touched, so no live bank token can outlive the account.
2. All vault credentials belonging to the member are deleted.
3. A household with no other enrolled member is deleted entirely, every table, in foreign-key order computed from the catalog.
4. In a household shared with others, the member's private records are deleted and their profile anonymised; jointly-owned ledger records remain with the other members.

5. The identity (sessions, factor, Google link, user row) is deleted.
6. One audit row records that the erasure happened, without personal data.

Requests by email are honoured within 7 days. Backup copies age out within 30 days. Tests cover both the sole-member and shared-household cases (`test/erase.test.ts`, `test/identity.test.ts`).

9. Privacy and consent

The privacy notice (`web/privacy.html`) is linked from sign-in and from the account panel. Consent to the current version is required at sign-in and recorded on the account with its version and timestamp (`app_user.consent_version`, `consent_at`). A member who signed in before a notice version is asked to agree on their next visit. No bank connection can be started without recorded consent.

10. Third parties

Provider	Purpose	Data sent
Google	Sign-in; cloud hosting	Email/profile at sign-in; all data at rest on encrypted disks
Plaid	Bank connections	Handled on Plaid's pages; Clan receives account/transaction data and a token
OpenRouter / Anthropic	Language model behind the agent	Message, recent conversation, household context needed to answer; never credentials or tokens. Accounts configured to exclude training on inputs.
Stripe	Subscription payment (hosted checkout)	None from Clan

11. Review and change log

Reviewed annually, on any architecture change, and after any incident.

Date	Change
2026-09-21	First version, written for Plaid production review. Adds TOTP second factor, recorded consent, account erasure, 2-day patch SLA, retention job, Dependabot and audit CI.