Affirmation Lock — App Privacy Summary
This document describes how **Affirmation Lock** handles information based on the current product implementation. It is meant to support **App Store Connect** privacy disclosures, **Google Play** Data safety, and an in-app or website privacy notice. **It is not legal advice**; have qualified counsel review before you publish or rely on it for compliance.
Last updated: April 9, 2026
App: Affirmation Lock (React Native; iOS and Android)
---
1. High-level principles
- **No traditional accounts in V1:** The app does not require you to sign up with an email or password. A **device-local subscription profile identifier** is generated on first use to tie subscription and optional server features together.
- **Core state is local-first:** Preferences, onboarding answers, selected apps to limit, affirmation text, journal cache, streak/journey data, and similar data are stored **on your device** (via encrypted-at-rest device storage used by the app).
- **Some features use our servers:** Optional **AI-generated affirmations**, **subscription verification**, and **product analytics** send limited data to services we operate or use as subprocessors (see below).
---
2. Data stored on your device
The app persists data locally so it can work offline and keep your routine private by default. Examples of what may be stored on-device include:
Category	Examples (non-exhaustive)
Preferences	Language, affirmation tone/category, analytics-related preference flags
Onboarding & setup	Steps completed, answers you gave during onboarding
App limiting (your choices)	Apps or categories you chose to protect, schedule windows (“daily protections”), and related status from the OS
Affirmations	Current and cached affirmation text, timestamps
Sessions	Temporary unlock sessions (timing, which apps were included in the session where applicable)
Journey / streak	Journey start date, missed days, and related keys used for the streak view
Subscription profile	A **randomly generated profile ID** (not your name or email) used to sync subscription state with our backend and paywall tooling
Paywall / subscription UI state	e.g. last paywall shown, placement names, last sync errors (for support/debugging in-app)
iOS: Limiting apps uses Apple’s **Screen Time / Family Controls** APIs. Identifiers for selected applications are handled according to Apple’s platform rules; the app does not need your messages, photos, or browsing history for this feature.
Android: Where the product uses **accessibility** or related services to detect foreground apps for gating, the app is designed to use that access **only to enforce the limits you configure**. In-product copy states that it does **not** read your typing, messages, or private content for that purpose.
---
3. Data sent to our services and partners
3.1 Product analytics (PostHog)
When analytics is configured in the app build, we use **PostHog** to understand how the product is used (e.g. funnels and stability), not to sell your data.
Typical metadata attached to analytics sessions may include: app name, platform (iOS/Android), and whether the build is development or production. The SDK may also capture **app lifecycle events** (e.g. foreground/background) when that option is enabled.
Events the app is built to record (names and properties may evolve in updates) include, for example:
Event (conceptual)	Example properties
Onboarding completed	Count of apps you chose to limit
App selection completed	Count of selected apps
Affirmation shown	Source of the affirmation (e.g. remote, cache, local), tone

Affirmation completed	Tone
Unlock session granted	Unlock duration, count of apps in session
Paywall shown	Paywall placement name, whether UI was presented
Accessibility disclosure (Android)	Platform marker
We do **not** intend for analytics to collect the full text of your affirmations as part of these events.
PostHog processing: Governed by PostHog's terms and privacy policy. Data is sent to the configured PostHog region/host (e.g. US instance when using `us.i.posthog.com`).

3.2 Paywall and subscription tooling (Superwall)
The app may integrate **Superwall** to show paywalls and experiments. Superwall is configured with a **public API key** and identifies the installation using the same **subscription profile ID** as other subscription features so offers and subscription state can stay consistent.
Superwall processing: Subject to Superwall's privacy policy and data processing terms.

3.3 Subscription and entitlement backend (Google Firebase)
We use **Firebase** (typically **Cloud Functions** and **Firestore**) to:
- **Verify and sync subscription state** between the app and our records, including when you purchase, restore, or when the store sends server notifications.
- **Gate optional remote AI** so that only entitled users receive server-generated affirmations.
Data the client may send to our HTTPS functions includes:
- **Subscription profile ID** (see above)
- **Platform** (`ios` or `android`)
- **Store snapshot** from the device, such as subscription status, product identifier, transaction identifiers, expiry/grace dates, and related fields needed to verify purchases (exact fields depend on platform APIs).
- **Paywall placement** name when syncing after a paywall flow
Server-side storage (Firestore): Subscription-related profiles and webhook event metadata may be stored to process renewals, cancellations, and entitlement checks.
Firebase processing: Subject to Google's Firebase terms and privacy documentation.

3.4 AI-generated affirmations (Firebase → model provider)
If remote generation is enabled and you are entitled, the app may **POST** a JSON payload to our **Cloud Function** proxy. That payload can include:
- **Locale** (language)
- **Tone** and **category** for the affirmation
- **Word limit** for the model
- **Display names of apps you selected** for limiting (used only as *context* in the prompt—e.g. "distractions related to these apps")
- **Subscription profile ID** so the server can confirm entitlement before calling the model
The Cloud Function may call a third-party **large language model API** (e.g. OpenAI) to generate text. **API keys for the model stay on the server**; the mobile app does not embed them.
We do not use this flow to train third-party models on your data unless a future version explicitly says otherwise and obtains appropriate consent; rely on your executed vendor agreements for the exact subprocessors and settings.
When remote generation is unavailable or you are not entitled, the app uses **cached or locally bundled** affirmations instead.
---
4. What we do *not* do (current design)
- **No login wall** with email/password in the described V1 scope.
- **No selling of personal information** as a business model for this document's purposes (describe any change if you add ad tech or data sales).
- **No direct OpenAI calls from the phone** for affirmation generation; requests go through our backend when enabled.
---
5. Legal bases and purposes (for GDPR-style disclosures)

Where GDPR applies, typical bases include **contract** (providing the subscription and features you asked for), **legitimate interests** (fraud prevention, abuse protection, product improvement via analytics), and **consent** where required (e.g. optional analytics or marketing—update if you add consent UI).
Purposes include: operating the service, verifying purchases, generating optional affirmations, improving the product, and protecting users and systems.
---
6. Retention
- **On-device data** remains until you delete the app or clear app data, unless the OS or backup features copy it (e.g. device backup).
- **Server-side subscription and webhook records** are retained as needed for accounting, disputes, and renewal processing—define specific periods in your internal policy.
- **Analytics** retention follows your PostHog project settings.
---
7. Security
We use industry-standard transport security (HTTPS) for calls to our backend. You should keep your device OS updated and use device passcodes/biometrics as appropriate.
---
8. Your choices and rights
Depending on your region, you may have rights to **access**, **correct**, **delete**, or **export** personal data, or to **object** to certain processing. Because many identifiers are **pseudonymous** (profile ID), you may need to provide proof of purchase or device information to exercise rights related to subscription records.
Analytics: The codebase includes a preference field for analytics; ensure your **Settings** UI and implementation match what you promise users (opt-out must actually disable collection when you ship it).
Subscriptions: Cancel or manage through **Apple App Store** or **Google Play** subscription settings.
---
9. Children
The app is not directed at children under 13 (or under 16 where applicable). Do not use the service for child-directed use cases unless you implement age gates and comply with COPPA/GDPR-K and store rules.
---
10. International transfers
If you use US-hosted vendors (PostHog US, Google Cloud, Superwall, OpenAI), data may be processed in the United States or other regions per those vendors' documentation. Use **Standard Contractual Clauses** or other mechanisms as required for your users.
---
11. Changes
When you change data practices, update this document, the in-app notice, and **App Store / Play Console** privacy answers before shipping the release.
---
12. Contact
Insert your legal entity name, address, and privacy contact email for your jurisdiction.

Appendix A — Apple App Store "App Privacy" (nutrition label) mapping
Use this as a checklist when filling **App Store Connect → App Privacy**. Adjust to match your **actual** shipped build and vendor contracts.
Data type	Collected?	Linked to user?	Used for tracking?	Notes
Purchase history (subscription)	Yes	Yes (profile ID)	No	StoreKit / Play Billing via your backend
User ID (pseudonymous profile ID)	Yes	Yes	No	Not email; device-generated
Product interaction (analytics events)	If PostHog on	Often "yes" in Apple's sense	Depends on PostHog config	Declare per Apple definitions
Other diagnostic	If any	…	…	Crash tools if added later
Apple's definitions of **"tracking"** and **"linked to you"** are specific; follow their current questionnaire wording.

Appendix B — Google Play "Data safety" hints

Declare: data types collected, whether collection is optional, purposes, encryption in transit, whether users can request deletion, etc. Align with sections 2–4 above.