← All press kits PRIVACY

Your Gift Cards Are Nobody Else's Business

A press-kit deep-dive on CardCue Pro's privacy architecture: a SwiftData-first, biometric-gated, App-Group-isolated wallet where authentication is optional, tracking is absent, and the only data that ever leaves your phone is the image you explicitly point at a gift card.

THE ONE-SENTENCE VERSION

The default CardCue Pro install stores every card on your device, hides secrets behind Face ID, ships zero third-party tracking SDKs, makes account sign-up optional, and sends only the single photo you scanned to one vendor (Anthropic) under a **Zero-Data-Retention** agreement.

~240 lines · SwiftData · Keychain · App Groups · LAContext

Read the white paper → Source markdown ↓

Everything stays inside the dashed line. The one exception is the scanned image, processed under Anthropic's Zero-Data-Retention agreement.

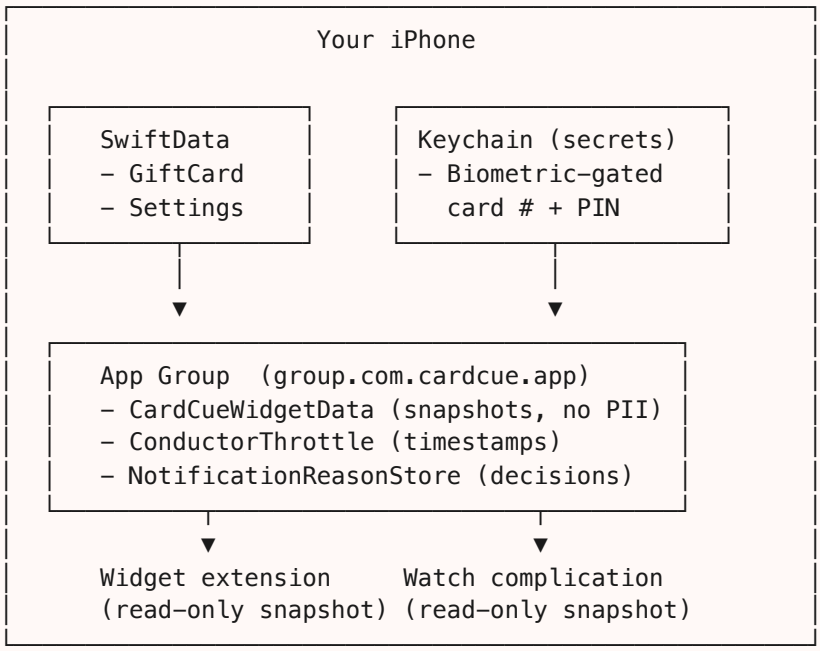
1. The premise

Gift-card data is unusual: it's mundane enough that people assume it's harmless, and sensitive enough that a leaked card number and PIN gets drained in minutes on a forum. Most wallet apps split the difference by copying it to the cloud "for sync" and calling it a feature.

CardCue Pro's architecture starts from the opposite assumption: *this data doesn't need to leave your phone for the app to work*.

There are now two ways to get a card into the wallet, the camera and the iOS Share Sheet from Mail, and both stay on-device by default.

2. Where your data lives



The only thing that ever leaves the phone:

> A single scanned image, on tap of the scanner, to Anthropic's API under their Zero Data Retention agreement. Image contains no user identity. Response is JSON, saved locally, never echoed back out.

WHAT IS NOT IN THE CLOUD BY DEFAULT

- Your wallet
- Your card balances
- Your card numbers or PINs
- Your scan history
- Your location
- Your notification decisions
- Your dismiss history
- Your quiet zones

3. Authentication is optional

CardCue Pro's Supabase-backed account layer exists for users who want backup and cross-device sync. It is **opt-in**. The entire core app, scanner, wallet, geofence notifications, widgets, watch app, live activity, CarPlay, works without ever creating an account.

When you do sign up, what gets synced is an *encrypted blob of card data* and the metadata the server needs to route sync messages (your user ID, a last-updated timestamp). The server cannot read your card numbers; they are encrypted client-side before upload.

When you delete your account, the server-side erase is initiated and the local SwiftData store is emptied. Account deletion does **not** require emailing support, it's a button in Settings.

---

4. Secrets are biometric-gated

A `GiftCard` has a `hasSecrets: Bool` flag set whenever a card number or PIN is present. Every view that renders those fields wraps the read in `LAContext`:

```
private func revealCardNumber() {
    let context = LAContext()
    context.evaluatePolicy(.deviceOwnerAuthenticationWithBiometrics,
                           localizedReason: "Show your card number") { success, _ in
        if success {
            DispatchQueue.main.async { self.numberIsVisible = true }
        }
    }
}
```

The SwiftData store itself is not encrypted at a per-field level, we rely on iOS's standard file-level encryption (Data Protection: *Complete Until First User Authentication*).

The biometric prompt is a *presentation* gate, not a storage gate. What this buys us: a shoulder-surfer who picks up your unlocked phone can browse your wallet list, but cannot reveal a card number without your face or fingerprint.

5. The App Group is read-only for extensions

Widgets, the Watch complication, and the background scan scheduler all read from `App Group: group.com.cardcue.app`. They **never write** to SwiftData. They publish timestamps (throttle state) and snapshots (`CardCueWidgetData`) through the App Group as a one-way channel from the main app outward.

This is important because widget extensions on iOS run in a restricted sandbox where a SwiftData container write can corrupt the main app's store. CardCue Pro avoids the risk entirely, the extension's only job is to read the JSON snapshot the main app wrote on its last refresh.

Card numbers and PINs are **not** in the widget snapshot. The widget can display balance, name, color, expiry, nothing that would be sensitive on a lock screen.

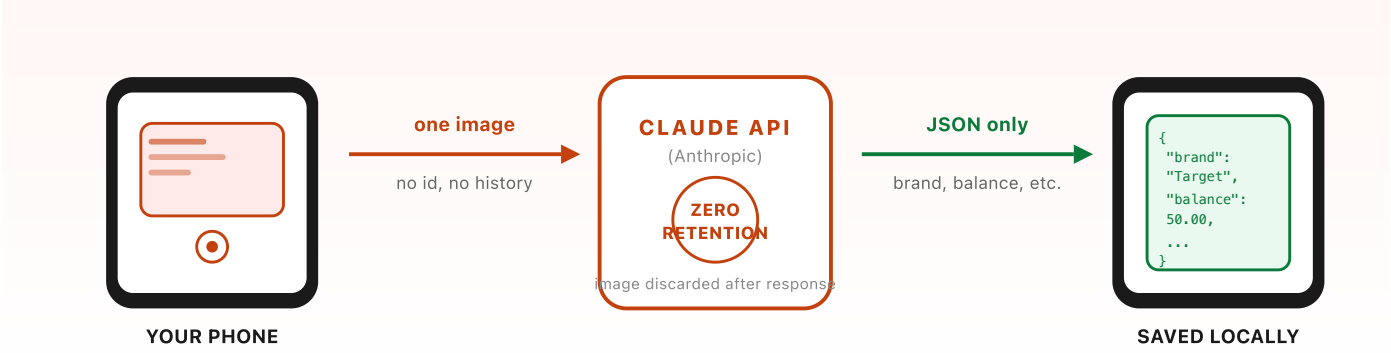
The same one-way App Group pattern now carries the new **iOS Share-from-Mail extension**. The extension writes a pending intake payload into `group.com.cardcue.app.shared` and the main app drains it on the next foreground. The payload is local, scoped, and ephemeral. If the app is killed mid-intake, the payload waits in the App Group until the next drain; it never leaves the device, and it never reaches any process besides the main app binary. (The user-facing Share flow itself is documented in the Scanner kit.)

The Share Extension target carries **no Claude API key**. The key lives in the main app binary only, because only the main app ever has a reason to call out. The extension's `Info.plist` has no key, the extension binary has no key, and there is no network code path inside the extension to use one. Parsing is done with *regex over the email text*,

text that is already on the user's device because Mail has already rendered it. The extension is, by construction, incapable of contacting Anthropic. It writes a local payload, and that is the extent of its reach.

THE SHARE EXTENSION'S PHYSICAL LIMITS

- No API key in the extension's `Info.plist` or binary.
- No network entitlement used by the extension.
- Parsing is regex over text Mail already rendered locally.
- The App Group is the only surface the extension writes to.
- The main app is the sole reader and drainer of that payload.



The scanner's round trip: one image out, structured JSON back, nothing kept on the other side.

6. What Anthropic (Claude) sees

When you use the AI scanner, CardCue Pro sends:

- The single card photo you just captured.
- A prompt asking for structured JSON extraction.

CardCue Pro does **not** send:

- Your name, email, or user ID.

- Your device identifier.
- Your location.
- Any other card in your wallet.
- Any usage history.

The request goes to Anthropic's Claude API under their **Zero Data Retention** commitment for business customers: the image is processed in-memory, the JSON response is returned, and the image is not stored, logged, or used for training. This is a contractual obligation, not a setting we hope they honor.

If you don't trust the cloud path, there is an app-wide setting "**Scan on-device only**" that disables the Claude engine entirely. With that enabled, every scan goes through `VisionOCRScanner` and never leaves the phone, at the cost of slightly less rich extraction for unusual cards.

---

7. No third-party tracking SDKs

CardCue Pro ships with:

- No Google Analytics.
- No Facebook SDK.
- No Adjust, AppsFlyer, Braze, Mixpanel, Amplitude.
- No crash reporter that phones home (we use Apple's built-in `MetricKit` for crash + hang reports, which stays in the Apple analytics pipeline the user controls via Settings → Privacy → Analytics).
- No advertising identifier (IDFA) access.
- No ATT prompt, because there's nothing to ask about.

Our `Info.plist` purpose strings are honest and specific, `NSLocationAlwaysAndWhenInUseUsageDescription` describes precisely what

background location powers (geofence nudges for nearby gift cards), not a vague "improve your experience."

8. App Store Privacy Nutrition Label

CardCue Pro's Privacy Label reflects the above:

Category	Status
Data Not Collected	Contact Info, Health & Fitness, Financial Info, Sensitive Info, Browsing History, Search History, Diagnostics, Purchases, Usage Data (when not signed in)
Data Linked to You *(opt-in, account-holders only)*	Email for account recovery; User ID for sync
Data Used to Track You	None

The corresponding metadata lives in `docs/AppPrivacyLabels.md` and is kept in sync with every App Store submission.

9. CCPA / LGPD / PIPEDA / APPI / POPIA / DPDP readiness

Requirement	CardCue Pro's implementation
Right to access / right to export	Settings → Export data produces a JSON of the user's entire wallet, settings, and reason log
Right to deletion	Settings → Delete Account triggers both local wipe and server-side erase

Requirement	CardCue Pro's implementation
Lawful basis	Performance of the contract (account-holders) and Legitimate Interest (location-based nudges, with clear opt-out)
Portuguese translation (LGPD)	In-flight for Brazil launch
App Store availability at 1.0	U.S., Canada, Australia, New Zealand, Japan, South Korea, Singapore, South Africa, India, Mexico, Brazil, and select Latin-American markets. **Not available in the EEA, the UK, or Switzerland at 1.0** pending the compliance work those jurisdictions require; re-evaluated post-launch.
Data Processor	Anthropic (Claude API). No other processors.

10. Files of record

File	Role
`Services/Services.swift`	Scanner pipeline, Claude call-site, on-device-only toggle
`Services/AuthService.swift`	Supabase auth, account deletion
`Models/Models.swift`	`GiftCard.hasSecrets`, SwiftData schema
`Views/Cards/CardDetailView.swift`	Biometric gate on reveal
`Models/Constants.swift`	`CardCueWidgetData` shared-snapshot shape
`docs/AppPrivacyLabels.md`	App Store Privacy Label source of truth
`Info.plist`	Purpose strings, entitlements

11. The stance

Privacy is a product feature, not a policy page. An app that asks for your camera, your location, your gift-card numbers, and your PINs has to earn that permission by proving, architecturally, that the data doesn't go anywhere it doesn't have to. CardCue Pro's architecture defaults to the shortest possible data path: from the card in your hand, to the phone in your pocket, and no further. That is the product's first promise, and the one the code is organized around keeping.

THE FIRST TIME

The first time a friend asks "which account did you sign in with?" and the honest answer is "none, it doesn't have accounts, nothing leaves my phone", and the app still knew the card, the store, and the moment, the privacy architecture stops being a compliance story. It becomes the shape of the product, the reason the right alert could arrive at all, and the reason the user trusted it enough to act on it in under thirty seconds.

Christian Sorensen, Founder · BigUnit Digital LLC
Read the founder's origin story →

CardCue Pro, by BigUnit Digital LLC. Built on SwiftData, Apple's platform encryption, LocalAuthentication, and a policy of not asking for data we don't need to deliver the feature.

A note on the writing. The thinking, the stories, and the product opinions are mine. I used AI to edit for grammar, tighten prose, and keep the voice consistent across the series. If a sentence lands cleanly, some of that credit goes to the machine. I figured you should know.

MORE PRESS KITS

COMPLETE KIT

CardCue Pro: The Whole App in One Read

ECOSYSTEM

The Intelligent Notification Ecosystem

SCANNER

Shutter-less Card Capture

GEOLOCATION

The Geolocation Brain

VOICE

The Voice at the Counter

FOLLOW CARDCUE PRO

X / Twitter Instagram TikTok YouTube Threads Mastodon RSS Email

 Back to press