

Split-Authority Custody: Hot and Cold Storage With Self-Recovery

Cove Wallet whitepaper

October 30, 2026 · @Matt

This paper describes Cove Wallet as built at walletd 0.51 and Cove 0.4.29 (October 2026), and marks what is planned but not yet built. Later changes are recorded in the sprint notes that accompany it, and features are added to this paper when they ship.

Abstract

A person can hold their own keys, spend from a phone every day, keep savings offline, and recover from losing the phone, the card or a written backup. They need no custodian, co-signing service or recovery company to do it. This paper describes how Cove Wallet does this on the XRP Ledger, its only chain at launch; four more are planned (Section 15). Responsibility is split across things a person already controls: a phone, a **Cove Card** (an NFC smart card), and two written-down recovery keys. Each is guarded by its own PIN or device credential.

The design rests on seven ideas:

- **Proven secure hardware, not a new device.** Instead of building a bespoke hardware wallet, with its own circuit board, firmware and supply chain to secure, Cove uses the secure hardware already in phones and the payment-grade secure element in an NFC card: the most heavily researched and tested secure chips there are.
- **Two accounts, two tiers.** Spend is a hot account the phone signs for, with its key sealed by the Secure Enclave (iOS) or Android Keystore. Save is a cold account the card signs for, inside its secure element and behind an 8–12 digit card PIN. They are separate ledger accounts, so the boundary between them is enforced by the ledger, not by the app.
- **Split authority.** Each key stays whole where it lives, and the XRP Ledger's native signer lists decide which combinations of keys can move each account. Keys are never copied or combined on a phone, during setup or recovery.
- **Two keys for savings.** Save needs two of three: the card, the phone and Save's recovery key. A stolen card and its PIN can't move it.
- **Self-recovery with what's left.** A lost phone is recovered with the card, which moves everything on the old phone's Spend to the new one, and with Save's recovery key it puts the new phone on Save. A lost card is recovered with the phone and Save's recovery key. Losing both is recovered with the two written-down keys.
- **A backup that checks itself.** Recovery keys are standard 24-word BIP-39 phrases. They are written as the Cove Key: 12 short rows of hexadecimal word indices, each with a check value that names the row a mistake is on.
- **A firewall in the signer.** Cove Shield blocks payments to addresses known to belong to attackers. The check runs inside the signing component on the phone, so a compromised app can't skip it.
- **What you approve is what is signed.** The signing component writes its own description of the exact bytes it is about to sign; the phone's system confirmation sheet, which no app can draw or change, shows those words; and only that sheet releases the key. A hacked app interface, the route behind the largest theft in the industry's history, can't get a different transaction past the user (Section 7).

Defences that hardware wallets leave to advanced settings, if they offer them at all (co-signing, a duress wallet, recovery from any single loss), are part of Cove's default, simple flow. The goal is self-custody that is **secure by default and simple to use.**

We explain why Cove builds on phones and cards, then describe the architecture, how keys are generated and stored, the backup format, signing and recovery, Cove Shield and the backend behind it, and the threat model, including the risks that remain.

1. The problem

Self-custody usually makes people pick two of convenience, security and recoverability. Exchange custody is convenient and recoverable, but it puts funds behind an intermediary. Hardware wallets are sold as the secure choice, but they are one more device to carry and lose, and a new computer with its own attack surface, less tested than the phone it was meant to replace (Section 3). Social and MPC recovery services restore access, but they bring a third party back in.

Three failure modes cause most real losses:

1. **A single point of compromise.** A seed phrase in a photo, a notes app or a drawer is a complete key. Whoever finds it owns the funds.
2. **Unrecoverable loss.** A lost device with no usable backup, or a backup copied down wrongly, ends access for good.
3. **Being tricked into paying an attacker.** The keys are safe, but the owner signs a payment to an address that belongs to a thief.

Cove addresses each one. No single object can move savings. Every loss of one thing, and the loss of both devices, is recoverable by the owner alone. The backup format catches transcription errors while the original is still on screen. Cove Shield refuses known attacker addresses before anything is signed.

2. Design principles

1. **No intermediaries in the key path.** No custodian, co-signing server, cloud key escrow or Cove-run recovery guardian (guardian keys, Section 13, are people the user chooses). Key generation, signing and recovery run on the user's devices. None of them depends on a Cove service being online.
2. **No single object moves savings.** The card needs its PIN. Without the card, Save needs two independent keys together.
3. **Tier by consequence.** Money needed quickly sits apart from money being saved. Fully compromising the hot tier must not reach the cold tier.
4. **Let the ledger enforce the rules.** Limits and signer rules live on the XRP Ledger, where a compromised phone can't change them.
5. **One component holds keys.** A single, small signing component, walletd, is the only code that ever touches a private key. Everything else, including the user interface, is treated as untrusted.
6. **Open standards.** Recovery keys are standard BIP-39 phrases, derived on the standard XRP path, so funds stay reachable with other wallets if Cove disappears.
7. **Reuse proven secure hardware.** Phones and payment-grade cards already carry the industry's best-tested secure chips. Cove builds on them rather than designing a new device (Section 3).
8. **Collect as little as possible.** Where a service needs a backend, it keeps only what it must (Section 11).

3. Why a phone and a card, not a bespoke device

A hardware wallet is a small computer built from scratch to keep one secret: a circuit board, a microcontroller (often a low-cost, general-purpose one, sometimes with a secure element beside it), a screen, buttons, a USB or Bluetooth link, a battery, a bootloader, firmware and an update channel. Each of those is part of the trusted computing base, and together they bring the same kinds of attack surface as a smartphone: firmware that can be replaced or updated, radio and USB links, and a screen usually driven by the general-purpose chip rather than the secure element, so what it shows can be altered or observed. But they get a fraction of a phone's scrutiny. The microcontrollers were never designed to resist physical attack, and the secure elements in wallets are mostly closed chips that haven't had the years of public review that phone and payment chips have. Each part also has to be manufactured and shipped without being tampered with. Calling such a device "secure" rests largely on its maker's word, and the public record shows how often that fails: researchers have extracted secrets from hardware wallets by voltage glitching and side channels (the "wallet.fail" talk in 2018, and Kraken Security Labs' glitching attack on a popular model in 2020), counterfeit and tampered devices have been sold to unsuspecting buyers, and in 2023 a vendor's optional seed-backup service showed users that firmware could export the secret it was meant to guard.

Phones and payment cards already solve this, at far greater scale. The secure hardware in phones (Apple's Secure Enclave, Google's Titan M and Android's StrongBox and TEE) and the secure elements in payment cards and passports carry billions of dollars of research, certification and attack-and-defence experience. They are built by the largest chip makers, tested by independent laboratories against published standards, and updated through mature supply chains. The banking system already trusts EMV payment chips with card secrets for billions of transactions a day. If the cryptography in those chips were broken, it would be broken for the banks too.

Cove uses those pieces instead of rebuilding them:

- **The phone** is the screen, the keyboard, the network connection and the everyday signer. Its secure hardware seals the hot key, and it is always with you and always updated.
- **The Cove Card,** an NFC smart card, holds the cold key in a payment-grade secure element. It has no battery, no screen, no ports and no firmware for the user to update: fewer moving parts and less to attack. It is powered only for the moment it is tapped, and only does anything with its PIN.
- **The XRP Ledger's signer lists** enforce who can move what, in the network itself, rather than in a device's firmware.

What a bespoke device rarely offers. Most losses don't come from the device at all, but from a lost device and a lost backup, a payment to the wrong address, or someone forcing the owner to unlock. Few devices defend against those, and where they do the defences are advanced settings: some devices offer a duress PIN or a hidden passphrase wallet, and multi-signature usually means coordinating extra software and several devices. Cove makes them **part of the default, simple flow**:

- **Self-recovery** of everything, from what's left after a loss (Section 9);
- **co-signing** as simple as scanning a QR code on a friend's phone (Section 4), with no extra software or second device to buy;
- **a duress wallet**, set up in a minute, that also locks the real wallet on that phone for three days (Section 10);
- **Cove Shield**, refusing known attacker addresses before anything is signed (Section 11);

- **two tiers** (spending and savings) so the amount at risk on the phone is the amount you chose to keep there.

The aim is self-custody that is **secure by default, and simple enough that the secure choice is the easy one.**

Cove's own cold-storage platform. Today's Cove Card is made for Cove by a payment-card manufacturer and runs that maker's applet. Cove will also publish its own example of a cold-storage platform on different hardware, starting with an open, BSD-licensed card applet (Section 8), so the whole stack can be inspected and reproduced.

4. Architecture

- **walletd** is a Rust component compiled into the app. It holds every key, decides what may be signed and builds each transaction itself. It links no networking code at all.
- **The app interface and network client are untrusted.** They can ask walletd to sign. walletd checks the request against its own policy and the user's contacts. The app supplies ledger facts (sequence number, fee, expiry), but those are bounded and can't redirect funds.
- **The card** talks to walletd over an encrypted session. The phone's NFC layer only relays opaque bytes. Every signature the card returns is checked against the card's known key before it is used. A mismatch stops the operation.

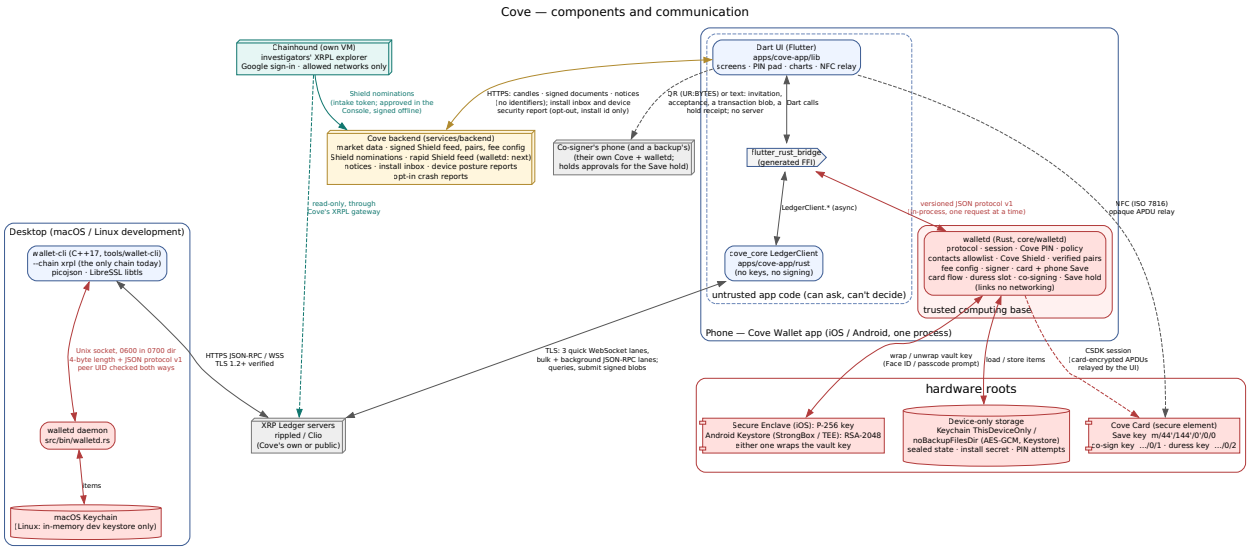


Figure 1. Components and how they talk. Red holds or protects keys; blue is untrusted Cove code; grey and yellow are outside the key path.

Component	Where	Trust
App interface (Flutter, `apps/cove-app`)	the phone	untrusted: it can ask
Ledger client (`cove_core`, Rust)	the phone	untrusted: network only, no keys
walletd (Rust, `core/walletd`)	inside the app; a daemon on desktop	**trusted**: it decides

Component	Where	Trust
wallet-cli (C++, `tools/wallet-cli`)	desktop	untrusted: it asks walletd over a local socket, checked both ways
Cove backend (Rust, `services/backend`)	Cove's servers	outside the key path (Section 11)

The components are named for the wallet, not the chain. walletd's custody, PIN, card and policy layers don't depend on the XRP Ledger, and wallet-cli picks its chain with `--chain` (today only `xrpl`), so each new network adds a codec, a key path and a command set rather than a new wallet.

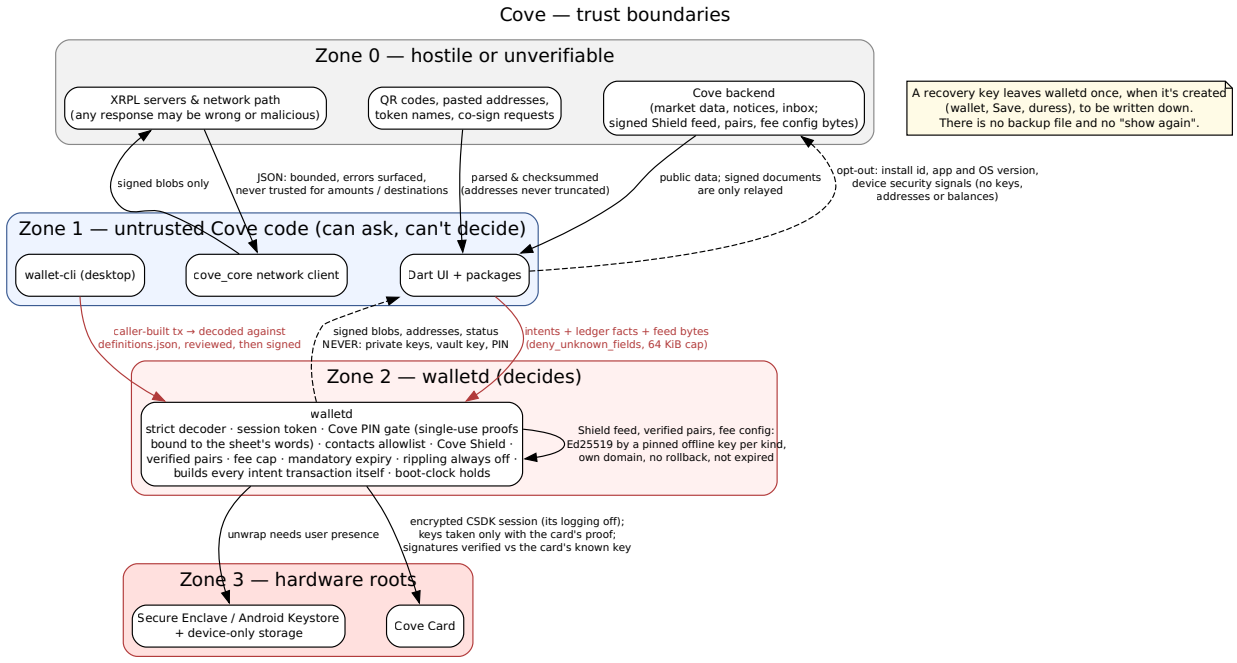


Figure 2. What crosses each trust boundary, and how walletd treats it.

Accounts and keys

Key	Lives in	Signs for	If it is compromised
Wallet key (secp256k1, from the wallet's 24-word recovery key)	the phone, sealed (Section 7)	Spend	the Spend balance
Save signer (secp256k1, from the same 24 words, its own path)	the phone, opened with the wallet key	Save, as one of its three signers	alone: nothing
Card keys (secp256k1, `m/44'/144'/0'/0/0` and `/1`)	the Cove Card's secure element	Save, as one of its three signers; Spend, as its rescuer	the Spend balance, with the card in hand and its PIN; nothing on Save alone
Save recovery key (a second 24-word phrase)	written down only	Save, as one of its three signers	alone: nothing
Card duress key (`m/44'/144'/0'/0/2`, optional)	the card's secure element	the duress wallet, as its rescuer	the duress balance, with the card and its PIN

On the ledger (Figure 3):

- **Save**'s own (master) key is off. Its signer list is {card: 1, the phone's Save signer: 1, Save recovery key: 1}, quorum 2: any two of the three, never one. Everyday, that's the card and the phone, signed in the same tap.
- **Spend** is signed by the wallet key. Its signer list is {the card's Save key: 1, the card's co-signing key: 1}, quorum 1, so the card can rescue Spend.

Why two accounts. The XRP Ledger can't limit how much one signer moves from one account. Keeping hot and cold funds in separate accounts makes the limit real on the ledger: a fully compromised phone can lose at most the Spend balance. Spend has no internal caps. It is the wallet in your pocket, holding what you chose to move there.

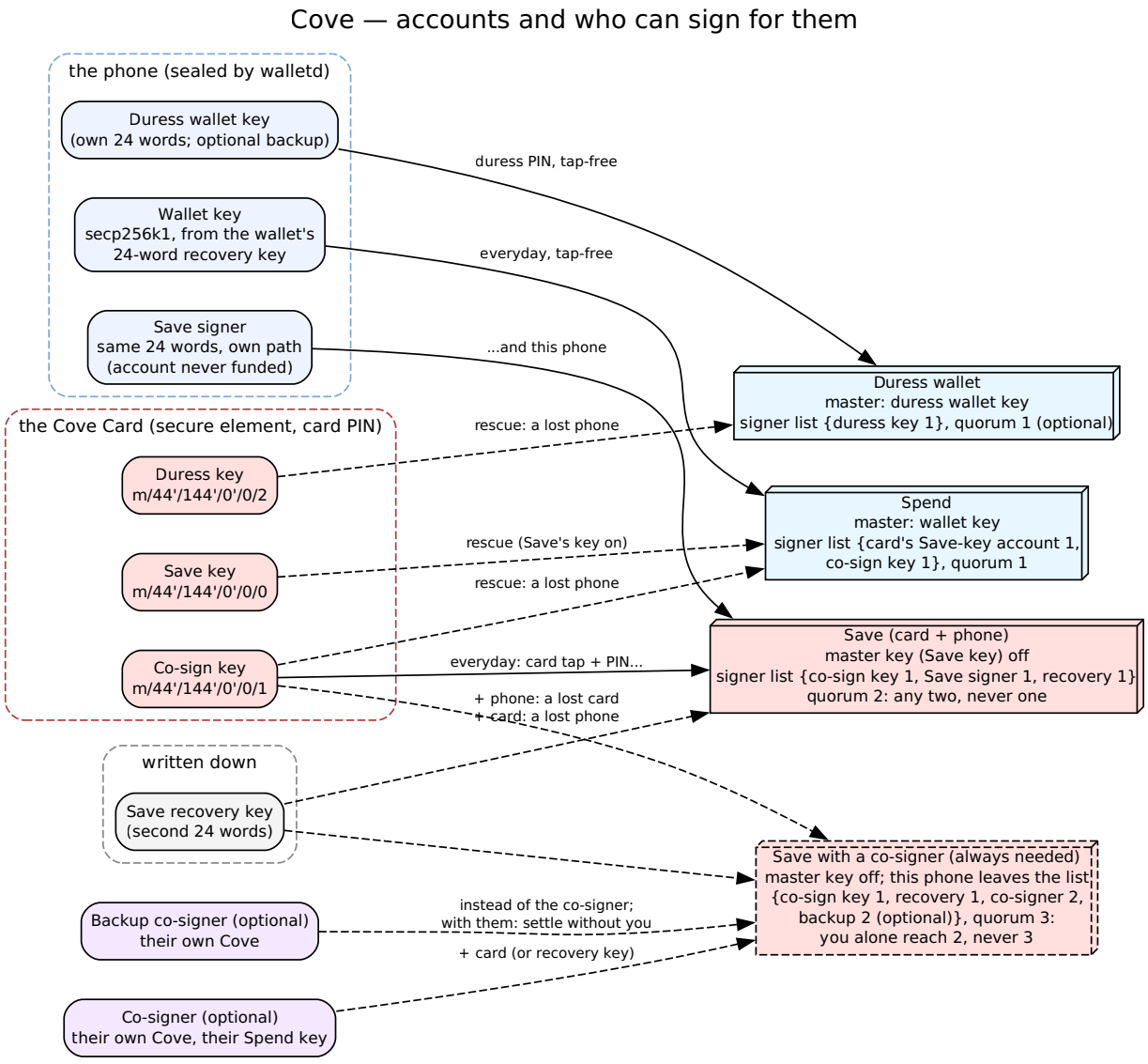


Figure 3. Accounts and who can sign for them. Solid lines are everyday signing; dashed lines are recovery and optional layers.

Optional layers

- **A co-signer on Save.** Save can require a second person: every Save transaction then needs the card and the co-signer, or the recovery key and the co-signer. The request moves between phones as a QR code in the open Uniform Resources format. No server is involved, and the co-signer's walletd checks every signature already on it (Figure 4). The card signs here with a second key of its own (`m/44'/144'/0'/0/1`), because a signer list can't name its own account. On by default, a **hold** makes each Save transaction wait days after the co-signer approves (below). A **backup co-signer** can approve instead when the co-signer can't, and the two co-signers together can **settle** Save without its owner, as an estate needs (Cove 0.4.6 and 0.4.7).
- **Backup cards.** A second (or third) Cove Card can be given the same Save key, so losing one card isn't losing Save. Once walletd pins Arculus's card key (Section 8), the copy goes only to a card that first proves it is genuine by signing a fresh challenge with that key.
- **A duress wallet.** A second PIN opens a separate, ordinary-looking wallet (Section 10).

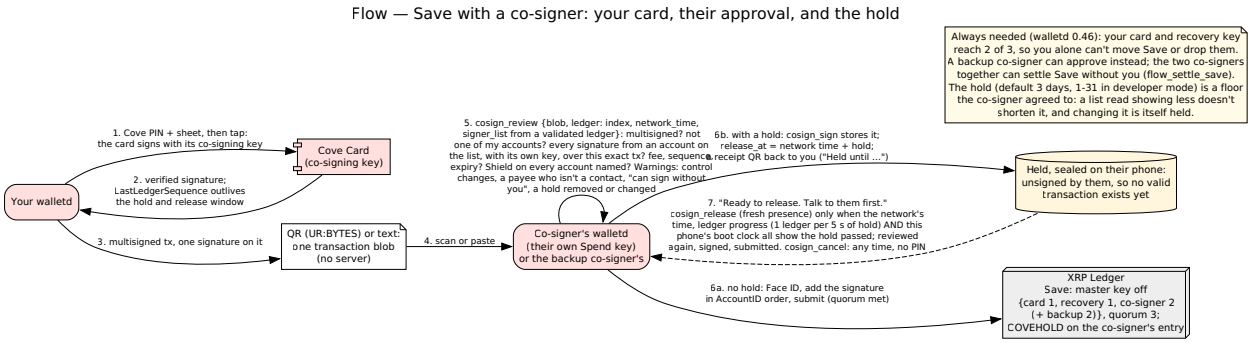


Figure 4. Save with a co-signer: the card on your phone, the approval on theirs, held for the agreed time before it's released.

How the ledger enforces it

Cove's rules about who can move what aren't settings in the app. They are three standard features of the XRP Ledger, which every validator checks on every transaction:

- **The master key.** An account's address is derived from a key pair, and that key signs for it. The account can turn its master key off (`AccountSet` with `asfDisableMaster`); the ledger then refuses any transaction signed by it.
- **The signer list.** An account can name up to 32 other accounts, each with a weight, and a quorum (`SignerListSet`). A transaction those signers sign together is valid when their weights add up to the quorum. A signer is named by its account, and signs with that account's master key or its regular key. The signer's account needn't hold any XRP, or even exist on the ledger: a key with no funded account behind it can still be a signer.
- **Multisigning.** A multisigned transaction has an empty `SigningPubKey` and a `Signers` array instead of one signature. Each signer signs the same transaction bytes with its own account ID appended (the `SMT\0` signing prefix), so a signature can't be moved to another transaction or another signer, and changing one field after anyone has signed breaks every signature on it. The fee grows with the number of signatures (the base fee × (1 + signers)), which walletd caps for each.

One card, three keys. The Cove Card makes its own seed in its secure element (Section 5) and derives three secp256k1 keys from it on the standard XRP path. None ever leaves the card; walletd reads only their public keys, and sends the card a 32-byte digest to sign.

Card key	Path	Its account is	Used as
Save key	`m/44'/144'/0'/0/0`	**Save** itself	Save's master key until Save's rules are set, then off; the signer on Spend's list
Co-signing key	`m/44'/144'/0'/0/1`	a signer-only account (Cove never funds it)	the card's place on Save's list; Spend's rescuer once Save's own key is off
Duress key	`m/44'/144'/0'/0/2`	a signer-only account (Cove never funds it)	the signer on the duress wallet's list

Separate keys are needed because a signer list can't name its own account, and so that the duress wallet's list doesn't name an account linked to the real wallet.

Save's rules, and how the card becomes Spend's rescuer. At card setup:

1. walletd reads the card's Save key and its co-signing key, in one session, and derives Save's address from the Save key. Save is the card's own account, so at first its master key is the card.
2. In one tap the card, still Save's master key, signs two transactions: Save's `SignerListSet` {card's co-signing account: 1, the phone's Save signer: 1, Save recovery key's account: 1}, quorum 2; then an `AccountSet` turning Save's master key off. From then on no single key moves Save.
3. The phone signs Spend's `SignerListSet`: {Save's account: 1, the card's co-signing account: 1}, quorum 1.

Each Save transaction is then built by walletd, signed by the card's co-signing key over its digest after the card PIN, and by the phone's Save signer, and verified before it leaves walletd: one tap, ready to submit.

Why the phone signs Save with its own key, not Spend's. The card can sign for Spend, so a thief with the card could set Spend's regular key, and a signer may sign with its regular key. If Save's list named Spend's account, the card would re-key Spend and then sign for Save twice over. So the phone's place on Save's list is a separate key from the same 24 words, at its own path (BIP-44 account `0x434F5645`), whose account is never funded and isn't anyone's signer. Restoring the 24 words brings it back. On Testnet, a thief holding every key the card has gets `tefMASTER_DISABLED` with Save's own key, `tefBAD_QUORUM` with the co-signing key alone, and `tefBAD_SIGNATURE` after re-keying Spend.

From setup on the card can sign for Spend as Spend's one signer: walletd builds a multisigned transaction on Spend, the card signs it with its co-signing key, and the ledger accepts it, with no involvement from the phone that made Spend. The ledger refuses a master key that's off as anyone's signer (`tefMASTER_DISABLED`, confirmed on Testnet), which is why Spend's list names the co-signing key. That is what makes recovery of a lost phone's Spend possible with the card alone (Section 9). The phone still signs Spend's everyday payments with its own master key; the signer list is an extra way in, not a replacement.

How co-signing works. Turning on a co-signer (a person Cove never contacts or holds a key for) changes Save's rules on the ledger in one card tap, after the Cove PIN and the system sheet naming the co-signer:

1. The card and the phone sign a new `SignerListSet` for Save:

Signers	Card (co-signing key)	Save recovery key	Co-signer	Quorum	Who can move Save
With a co-signer	1	1	2	3	the card + the co-signer; card lost: the recovery key + the co-signer

Signers	Card (co-signing key)	Save recovery key	Co-signer	Quorum	Who can move Save
… and a backup co-signer (weight 2)	1	1	2	3	as above, with either co-signer; the two co-signers together, without the owner

The co-signer is always needed: the card and the recovery key together can't move Save, or remove the co-signer, without them.

2. The phone leaves Save's list: with it on, the phone and the recovery key would be the owner alone again.

walletd treats the co-signer as live only once the app sees the new list validate on the ledger. Each Save payment is then:

1. built by walletd, which checks the destination (the user's own accounts and saved contacts) and Cove Shield;
2. signed by the card's co-signing key as a multisigned transaction carrying one signature, after the card PIN;
3. handed to the co-signer as a QR code in the open Uniform Resources format (or as text). No server sits between them; nothing in transit needs trusting, because every signature covers the exact bytes;
4. checked by the co-signer's own walletd: it must decode as a multisigned transaction for someone else's account; every signature already on it must verify over these bytes (a changed amount breaks the card's signature, and is refused); its fee must be capped and it must expire; and no account it names may be blocked by Cove Shield;
5. described by the co-signer's walletd in its own words, on their screen and in their phone's system sheet, with a warning for anything that changes who controls the account (signer lists, the master key, a regular key, deleting the account) or pays someone they haven't saved;
6. signed by the co-signer and handed back, ready to submit once the quorum is met.

Removing the co-signer needs their agreement: the card signs its share of Save's card + phone list and sends it to them as a request; once it validates the co-signer is gone and Save's own key stays off.

The Save hold. Accounts of physical attacks on crypto holders keep naming two things that stopped thefts: multisig, and withdrawals that take time. The co-signer is the first. The hold is the second: every transaction out of Save waits a set time after the co-signer approves it, and can be cancelled until then. A coercer needs the money to arrive while they still control the victim; the hold takes that away, and someone who escapes stops the transfer with one call to their co-signer, from any phone.

- **A choice when adding a co-signer, on by default.** Three days; one to 31 in developer mode. The co-signer agrees through QR codes between the phones: the owner's invitation states the terms, and the co-signer's walletd answers with an acceptance signed by their key, which the owner's walletd checks before the card signs anything.
- **On the ledger.** The card writes the hold into the co-signer's entry of Save's signer list, in the entry's free 256-bit `WalletLocator` field: `COVEHOLD`, a version, then the hold in seconds. It survives the co-signer's phone, anyone can read it, and changing it is a Save transaction like any other: turning it off or shortening it waits the hold too.
- **Enforced by the co-signer's walletd.** Approving doesn't sign. walletd keeps the request, unsigned, until the hold has passed three ways: by the network's time (a validated ledger's close time), by the ledger's progress since approval (at least one ledger per five seconds of hold), and by the co-signer's

phone's own boot clock, which a moved date can't change and a restart only restarts (walletd 0.47). One server's word can't end a hold, and nor can the phone's settable clock. Until then no transaction exists with enough signatures to be valid, so there is nothing to steal or submit early. Then the co-signer has seven days to release it (Cove PIN and system sheet, after "Talk to [owner] first"), and their phone submits it. Cancelling needs no PIN: it only stops money moving, and a cancelled request can't be approved again.

- **Requests last long enough.** walletd stretches a held Save request's expiry past the hold and the release window.

Every step was run against the XRP Ledger Testnet: the signer list with `COVEHOLD` on the co-signer's entry, a payment refused early, then released and validated, a cancel, and turning the hold off. Its limits: without a co-signer there's no hold, since the card and the phone sign at once; and a co-signer who releases without checking, or who is coerced too, gives the hold up.

5. Key generation

- **The wallet key and Save's recovery key** each start from 256 bits of the operating system's cryptographic random number generator. That generator is seeded by the platform's hardware sources. **The user then makes it theirs:** while the new key is on screen they can tap any word to bump it. walletd keeps a second, hidden 256-bit pool from the same generator, hashes the tap's microsecond timing and position into it (`pool = SHA-256("cove.entropy.v2" ‖ pool ‖ word ‖ length ‖ input)`), and re-rolls that one word from the pool; the rest of the key stays still (word 24's checksum bits follow). The user never chooses a value, so a bump can't make the key weaker or guessable, and every bumped word carries both the hardware's randomness and the user's: the key no longer rests on the hardware generator alone. **Set my key** then fixes it, and it's shown again if the user goes back. The resulting 256 bits become a 24-word BIP-39 phrase (256 bits plus an 8-bit checksum, as 24 words of 11 bits). The phrase is stretched with PBKDF2-HMAC-SHA512 into a BIP-32 seed. The account key is derived at the standard XRP path `m/44'/144'/0'/0/0`, so the same words open the same account in other XRP wallets. No BIP-39 passphrase is used.
- **The card key** is created by the card itself, from the secure element's own random number generator, when the card is set up. The card returns its phrase only at that moment. By default walletd discards it at once: Save doesn't need it, because its signer list can move the funds without the card. Users who want a card backup can choose to keep it. They then write it down once, as a Cove Key, and walletd checks it derives exactly the card's key before showing it. The app calls it **Save's own key**. It is the card's keys, so it counts as the card: with the phone or Save's recovery key it moves Save; alone it doesn't.
- **Two keys at Save setup, labelled apart.** Save setup can show two Cove Keys in a row: Save's own key ("KEY 1 OF 2", recommended) and the Save recovery key ("KEY 2 OF 2"), a separate key made by the phone that can move Save only together with the phone or the card (Section 9). Writing down Save's own key is recommended even with a backup card (three keys, each kept apart: cards can be lost together), and setup ends by listing what the user now holds and what each can do on its own, so the two are never mistaken for each other.
- **Signing.** XRP Ledger signatures use secp256k1 ECDSA with deterministic nonces (RFC 6979) and low-S normalisation. Every signature walletd or the card produces is verified before it leaves walletd.

6. The Cove Key: a backup that checks itself

Each of the 2,048 BIP-39 words is an 11-bit index, which is exactly three hexadecimal digits, from `000` ("abandon") to `7FF` ("zoo"). The Cove Key writes a 24-word phrase as 12 rows. Each row holds two words and two check digits:

```
row  1:  000 7FF  3C
...
row 12:  word23 word24  check
```

- **Check digits** are two Reed-Solomon parity symbols over GF(16), one per hex character, of the row number and the row's six word digits. The code's minimum distance is 3, so any one or two wrong characters, any two characters swapped and any row in the wrong place are caught: the most a two-character check can guarantee. The app says which row is wrong, as the row is typed, rather than BIP-39's "the phrase is wrong somewhere". The check digits carry no secret. The standard BIP-39 checksum is still verified on top.
- **Shorter and simpler to stamp:** 96 characters (72 word digits and 24 check digits) against about 130 letters for the full words, using only `0–9` and `A–F`, 16 punches instead of 26. Every row has the same fixed width, so a missing character is obvious.
- **A lookup table, not a cipher.** Anyone with the BIP-39 English list can turn a Cove Key back into the 24 words by hand. A backup never depends on Cove existing. The 24 words are shown as well, and restore also accepts them, including the four-letter abbreviations steel plates use.
- **Shown once, then confirmed.** A recovery key is shown exactly once, when it is created. The user proves they wrote it down by re-typing three rows walletd picks (or those rows' six words, for someone who wrote the 24 words), or the whole key with **Verify complete key**. walletd never shows it again. There is no backup file to export, because the written-down key is the backup.
- **Paper or metal.** The format suits a stamped steel plate, which survives fire and flood far better than paper.

How well it catches mistakes. One or two wrong characters in a row, any two characters swapped, or a row in the wrong place: always caught, by that row's check (proved over every one- and two-character error). Three or more mistakes in one row get past its check about 1 time in 236, and the BIP-39 checksum then catches nearly all of those, so about 1 in 60,000 such rows gets past both. That key restores a different, valid and almost certainly empty wallet: no funds are lost, but the user is confused. A word's first digit of `8–F` is always refused. (Earlier formats used a truncated SHA-256 and then a CRC-8, which missed some two-character errors; keys written with them restore from their 24 words.)

What the confirmation proves. Re-typing three of the twelve rows proves the key is in front of the user. It doesn't prove every row was copied right: one miscopied row has a one-in-four chance of being among the three asked. **Verify complete key** closes that gap by checking every row (or word) against the key, 1:1; it is meant to be the default in production. Without it, the row checks are the backstop: at restore, a miscopied row is caught and named.

The check digits are part of the secret. On a complete key they add nothing: they are a public function of the words, so the key is exactly as strong as the 24 words (256 bits of entropy behind a secp256k1 key, about 128-bit security). On a partial key they matter, because each row's check narrows that row's 22 bits to about 14. Three rows hidden from an attacker are 2^{58} guesses with the words alone, but 2^{34} if those rows' check digits are visible. So:

- protect the whole sheet: hiding "just the words" is not enough;
- **never split a Cove Key**, or any seed phrase, into pieces kept in different places. Half a phrase is 2^{124} guesses, and half a Cove Key plus the other half's checks is 2^{76}. Neither is a safe way to share a secret. Cove splits authority with signer lists instead (Section 4).

Typed only into Cove. Recovery keys are entered on Cove's own keyboard, never the system one, so no input method (a common Android keylogger) sees a keystroke. Rows take only hex digits, words only BIP-39 words, the typed characters stay out of the accessibility tree, and the screens block screenshots and screen recording.

Limits. English word list only: the same digits name different words in another language's list. Cove generates only 24-word keys; a wallet restored from another wallet's 12-word phrase works but has no Cove Key. No BIP-39 passphrase is used, so the 24 words alone open the account. The full design, proofs and test vectors are in `security/cove-key.md`.

Keep the two recovery keys apart. The wallet recovery key and Save's recovery key are independent, but together they meet Save's quorum. Anyone who finds both, stored together, can empty Save without the card.

7. The hot tier: the phone

Neither the Secure Enclave nor Android's Keystore can sign with secp256k1: they support P-256 and AES. So Cove uses the phone's secure hardware to

wrap

the wallet's keys, not to hold them as signing keys.

The sealed vault. walletd keeps the wallet's keys, contacts, Shield list and settings in one encrypted state. Opening it needs four things together:

- a 32-byte vault key that is only ever stored wrapped by the phone's secure hardware:
  - **iOS:** ECIES under a Secure Enclave P-256 key that requires Face ID, Touch ID or the passcode, and never leaves this device.
  - **Android:** RSA-OAEP under a Keystore key in StrongBox where available, otherwise the TEE, usable only inside a system biometric or screen-lock prompt;
- a random install secret, also kept in device-only storage;
- the device model, mixed in as context;
- the user's **Cove PIN**, stretched with Argon2id (64 MiB, 3 passes).

These combine (HKDF-SHA256) into the AES-256-GCM key that seals the state.

Secure hardware or nothing. A phone's keys only ever live behind its secure hardware: the Secure Enclave, or Android's StrongBox or TEE. If an Android phone's keystore turns out to be software only (an emulator, or a phone modified to fake it), release builds of walletd refuse to make or open a wallet on it at all. walletd decides this from its own build, so app code can't switch it off; only development builds allow it, for emulators.

How safe is this phone? About rates the phone **Secure**, **Warning** or **Insecure**, and this is the only place Cove warns about the phone. Insecure means the key could be exposed to other apps: no secure hardware, no screen lock, an unlocked bootloader or unverified system (from Android's key attestation), or a copy of the app not signed by Cove. A store build won't open the wallet on an Insecure phone; development builds still run, for testing. Warning means Cove is used outside its supported way: developer mode (unlocked by ten taps on About) has ever been on, or a debug or staging build. It is one-way: turning developer mode off doesn't clear it; erasing Cove and starting again (with a new key, or the old one recovered) does.

Properties:

- **The Cove PIN is never stored, not even as a hash.** The only way to test a guess is a full unlock, which needs the secure hardware and the user's presence. A copied backup or file dump offers nothing to brute-force offline.
- **Nothing migrates.** The keys are device-only, excluded from backups and cloud keychains, and can't be exported. A copy of every stored byte is useless on another phone.
- **Wrong PINs are counted, then the wallet is erased.** The phone's lock comes first: the PIN is never stored, only used with what the secure hardware releases after the phone's lock, so nothing can test a Cove PIN without it. Each guess is counted before it's tested (an app killed mid-guess doesn't get it free). The first five are free; then waits of 1, 5, 15 and 60 minutes and 3 hours, each saying how many tries are left; the **10th wrong PIN erases the wallet from the phone**: Spend's key, the duress wallet and Save's details. Nothing on the ledger is touched, and the recovery keys or the Cove Card restore it. Burning the count needs the phone's lock each time, so the erase can't be triggered by someone who only holds the phone.
- **Sensitive actions ask again.** Adding a contact, unblocking an address, showing a recovery key or erasing needs a fresh PIN and device check. Each check is good for one action. The phone's own confirmation sheet (drawn by Android or iOS, which no app can change) shows walletd's description of that action, and the app shows the same words with the PIN pad, warning if they don't name the address it asked for.
- **The Spend key locks when it isn't used.** Spend's secret is stored sealed under a key derived from the vault key, which only the system sheet releases, so with the key locked nothing in memory can rebuild it. Unlocking or entering the PIN opens it; quick payments and short trades then need no prompt. It locks after two minutes without a Spend signature, and ten minutes after it opened at the latest. The next signature asks for the PIN, and the system sheet shows walletd's own description of the exact transaction, decoded from the bytes to be signed ("Send 25 XRP from Spending to r… · fee …"; a token is named with its issuer). On iOS, Face ID doesn't display that text; the passcode screen does.
- **Secrets are wiped from memory** after use, kept out of swap where the platform allows, and core dumps are turned off.
- **The app locks** when you leave it (immediately by default) and after an idle time. It hides itself in the app switcher, and Android blocks screenshots of any screen showing a secret. On Android 12 and later, no other app may draw over Cove.

Who Spend can pay. walletd pays only the user's own accounts and saved contacts. Adding a contact needs the PIN, and every contact is checked against Cove Shield. The one exception is a **scanned payment request**: a shop's or an invoice's QR code (`xrpl:r…?amount=…&dt=…`), in XRP or a token Cove has reviewed, is paid to that address without saving it as a contact, but every such payment needs the PIN and the system sheet, even while the Spend key is open, and Cove Shield still applies. walletd builds each transaction from the request itself (who, how much), caps the fee, and requires an expiry, so a signed transaction can't be held back and used much later.

Trust lines can't drain an account. To hold a token on the XRP Ledger an account sets a trust line to its issuer. Only the account's own signed transaction can create one, and a line moves nothing by itself: XRP or tokens leave only in a transaction the owner signs. What a line can do is subtler:

- **Rippling.** If rippling is on for two lines of the same currency from different issuers, anyone can route a payment through the account, swapping its balance of one issuer's token for the other's at face value: a drain without the owner signing anything. Cove keeps rippling off on every line, and walletd enforces it in every signing path: it sets "no rippling" on each line it signs, and refuses to turn rippling on or to make it the account's default. If another wallet using the same key turned it on, the app shows it and turns it off in one tap.
- **Reserve.** Each line holds a small XRP reserve until it's removed. Removing a token returns any leftover too small to sell to its issuer (walletd fixes the destination and caps the amount) and deletes the line, then checks the ledger: if the issuer's own settings keep the line, the app shows that it's still there.
- **The issuer's powers over its own token.** An issuer can freeze its token, and if it enabled clawback, take its token back. That reaches only that token, never XRP or other tokens. Unverified issuers are hidden unless the user chooses to see them, and are checked by Cove Shield.

What you approve is what is signed

A common objection to phone-based wallets is that "it's just a phone, and nothing can help a hacked phone." That objection covers two different attacks. One is malware that takes over the phone's operating system itself. The other, far more common and far more costly, is a compromised app or signing interface: code that shows the user one transaction and sends another to be signed.

The second is how the largest theft in the industry's history happened. In February 2025 attackers took about $1.5 billion in ether from the exchange Bybit. They didn't break a key or a hardware wallet. They changed the web interface Bybit's signers used to approve transactions: it showed them a routine transfer, they approved it on hardware devices that showed only data they couldn't read, and the transaction they signed handed control of the wallet to the attackers. Every key worked exactly as designed; what failed was the link between what the signers saw and what they signed.

Cove is built so that link can't be broken from the app. Three parts have to agree before a key signs:

1. **walletd's words, from the bytes.** walletd, not the app interface, builds each transaction, or decodes one the app built, and writes its own description of it from the exact bytes it is about to sign: the type, the amount, the token with its issuer, every account it names in full, and the fee ("Send 25 XRP from Spending to r… · fee 0.000012 XRP"). It never repeats a description the app gives it.
2. **The system sheet, which releases the key.** Those words are shown in the phone's own confirmation sheet: Android's biometric prompt, or iOS's Face ID, Touch ID and passcode sheet. The operating system draws it; no app can change its text or draw over it (Cove also blocks overlays on Android 12 and later). Code can't skip the sheet: the vault key that opens the wallet is held by the phone's secure

hardware, which releases it only when the user confirms that sheet. Each approval is bound to the one request it was shown for (a digest of the command and its arguments), is used up by it, and expires after two minutes, so an approval can't be replayed for something else.

3. **The app checks walletd.** The Cove app shows the same words with the PIN pad, and independently compares them with what it asked for. If walletd's description doesn't name the address the app requested, the app replaces it with a red warning ("This isn't what Cove showed you. Don't confirm.") before the sheet appears.

A changed app interface can therefore lie on its own screens, but the words in the system sheet come from walletd and describe the real transaction. To send money somewhere else, it would have to get the user to approve walletd's description naming that somewhere else.

Cove doesn't rely on the user reading carefully, either. The rules below hold whatever the user approves:

- **Only known destinations.** Spend and Save pay only the user's own accounts and saved contacts. Adding a contact is itself confirmed in the system sheet, which shows the full address being added, and every contact is checked against Cove Shield. A scanned payment request is the one exception, and it is confirmed in the system sheet every time.
- **No hidden control changes.** The general signing path refuses every transaction that changes who controls an account: a new signer list, a regular key, turning the master key off, deleting the account, delegating permissions, or batches that hide inner transactions. Those happen only in Cove's own setup flows, each confirmed by name.
- **Nothing opaque to sign.** The Bybit transaction was a smart-contract call whose meaning wasn't visible in what the signers' devices showed. The XRP Ledger has no such calls: every transaction is one of a fixed set of types with typed fields, and walletd decodes every field against the ledger's own definitions before signing. There is no calldata it can't read.
- **A second person checks too.** With a co-signer, their own walletd decodes the transaction independently, verifies the card's signature over those exact bytes, describes it on their phone, and warns about anything that changes control of the account (Section 4).

What remains. Malware that controls the operating system itself, past its secure boot and with root privileges, could draw a fake sheet. That is why the About rating treats an unlocked bootloader, an unverified system or a missing screen lock as Insecure, and a store build won't open the wallet on such a phone; jailbreak and root detection and platform attestation are on the roadmap (Section 15). Even then, the ledger limits what such a phone can lose: Spend's balance, while Save needs the card tapped with its PIN (and then pays only the user's own accounts and saved contacts), or a recovery key that isn't on the phone. Two smaller gaps: on iOS, the Face ID sheet doesn't display the text (the passcode sheet does, and the app shows it beside the PIN pad); and while the Spend key is open after a recent approval (two minutes idle, ten at most), Spend's everyday transactions, such as payments to existing contacts and trades, don't ask again.

A hacked app therefore can't redirect a payment the user approves, can't add a payee the user didn't see named, and can't reach Save.

8. The cold tier: the Cove Card

Save is signed by the **Cove Card**: a contactless smart card with no battery, reached only over NFC at a few centimetres. The Cove Card is Cove's own; what makes it a Cove Card is the secure element and the applet it runs, not the plastic, so it can come as a branded card or as plain white plastic. Today's Cove Card is white-labelled for Cove by Arculus, a payment-card manufacturer, and runs Arculus's wallet applet, which Cove drives through Arculus's SDK. Arculus describes that card as Common Criteria EAL6+ certified; the claim is the maker's, and covers the card as the maker ships it. The card does nothing until the phone powers it with a tap and the right PIN is given.

- **Card PIN.** New card PINs are 8 to 12 digits. The guesses people try first are refused, such as repeated digits and runs like `12345678`. The PIN is checked on the card, which enforces its own wrong-PIN limit and reports the tries left after each miss. It travels to the card inside the encrypted session and is never stored. **How many wrong PINs the card allows, and what it does then, is not yet known.** We expect it to lock, at least for a time, after about ten. Testing this may destroy the card, so it is done last, on a spare card, before production. Either way the card can be reset (wiped) by anyone without its PIN, and a reset does not restore the tries, so a thief can destroy a card but not guess around the limit. A wiped card costs nothing that Save's recovery key and the phone can't recover.
- **Signing.** walletd builds the transaction and shows it. When the user taps, the card signs the digest with the Save key and returns only the signature. walletd verifies it against the card's known key before anything is submitted. The Save key never leaves the card.
- **Many transactions, one tap.** Adding Save's trust lines, or a whole recovery, is signed in one NFC session, up to 40 transactions at a time.
- **Is it a genuine card?** Every card setup starts with a check that the card is a genuine Cove Card, and **Settings → Save & card → Verify Card** runs the same check on any card, any time (one tap, no PIN, nothing changed). The card signs a fresh random challenge with a key held in its secure element, and walletd checks the signature against Arculus's card key, which Cove holds. The challenge is new each time, so an old answer can't be replayed: a valid signature proves that the chip in the field holds the maker's key. Cove confirmed the key with Arculus on two cards, and the card that signed for that confirmation came directly from Arculus. Arculus's integrator key is not needed for this check.
- **Two kinds of Cove Card.** An **Arculus card** is a certified secure element running Arculus's applet, proven genuine as above. Once it ships, a **Cove Technology open card** runs the same cryptography on an open BSD-3 applet, which anyone can read and verify, and which for that reason can't carry a certification. Anything else is reported as not genuine. Chip diagnostics are a developer tool (Card Info). walletd 0.51 recognises an Arculus card but doesn't yet check its signature against the key; the next walletd release pins the key and makes the check.

The display problem. The card has no screen, so the phone is the only display. A compromised phone could show one thing and ask the card to sign another. The card signs a digest; it does not parse transactions. Cove narrows this risk: walletd, not the app interface, builds every transaction, and Save's payments are limited to the user's own accounts and to saved contacts, each contact approved by its full address in the system sheet (Section 7). The worst a lying phone can do is send Save's funds to one of those. It cannot close the gap fully, and Section 12 treats it as a residual risk.

Why trust today's card. The current applet and its PIN handling are the card maker's closed code, and Cove has no independent penetration test of them yet. As far as we know, neither can be read back off a card, so part of their protection comes from not being open to inspection rather than from public review. Cove relies on the maker for three other reasons. Its cards come out of a payment-card factory that can

vouch for each card's integrity and its cryptographic features. Each card proves, with the maker's key, that it came from that factory. And the secure-element chips are the same kind used in EMV bank cards and other payment applications, which auditors and payment standards have evaluated. If the cryptography in those chips were broken, it would be broken for the banks too. An independent assessment of the card is on the roadmap.

Roadmap: a Cove applet. Cove will write its own card applet and publish it under the BSD 3-Clause license, as a reproducibly built CAP file with its own attestation key, so anyone can read the code and confirm what the card runs, and lets the Cove Card run on any compatible secure-element card. It will **not** be EAL6+ certified: the EAL6+ claim belongs to the maker's applet and SDK, not to code Cove writes. The open applet gives up the certification so that anyone can check the code instead.

9. Self-recovery

The states behind this section (a phone's wallet, Save on a phone, a Cove Card and the keys on paper) are set out step by step in `architecture/state-machines.md`.

Every loss below is recovered on the user's own devices. No support ticket, server or third party is involved.

Lost	What the user needs	What happens
Phone	the card and its PIN, and Save's recovery key	A new phone makes its own new wallet key. The card links Save and moves everything on the old Spend here (below); the card and Save's recovery key put this phone on Save's signer list in the lost one's place. The card then becomes the new Spend's rescuer. Without Save's recovery key, the card still brings back Spend, and Save waits for that key or the wallet recovery key.
Phone (alternative)	the wallet recovery key	Restoring the 24 words brings back the same Spend account, and with it the phone's place on Save's list (its Save signer comes from the same words). Cove finds Save on the ledger (Spend's signer list names the card's account, which is Save) and shows its balance, read-only; a tap of the card links it again.
Card	the phone and Save's recovery key	The phone's Save signer and the recovery key together meet Save's quorum and move its funds. A new card is then set up.
Phone and card	both written-down keys	The wallet recovery key restores Spend on a new phone. Together with Save's recovery key it meets Save's quorum.
Card and Save's recovery key	—	Save is lost for good: the phone alone is below Save's quorum. With guardian keys (Section 13), the two guardians together move it.
Phone and Save's recovery key (wallet recovery key also gone)	—	The card brings back Spend, but alone is below Save's quorum: Save waits until one of those keys turns up. That is the price of a stolen card being useless on Save. With guardian keys (Section 13), the card and a guardian move it.
A written-down key	the phone and the card	Nothing is at risk. Set up Save again to issue a new recovery key.

Recovery happens on the ledger. None of these recoveries asks Cove, or anyone, for permission. Each is a set of ordinary transactions that the signer lists set up in Section 4 already allow, so the ledger's validators are what accept them:

- **Lost phone, with the card.** Spend's signer list names the card, so the card can sign for the lost Spend. Save's list names the lost phone's Save signer, which must come off. The card's co-signing key and Save's recovery key (typed in, checked against Save's list on the ledger, used, then wiped) multisign a new `SignerListSet` for Save, naming this phone's Save signer instead: two of three, the quorum. From that ledger close on, the lost phone can no longer help move Save, even if it is found and unlocked. The card then multisigns, as the old Spend's one signer, the transactions that empty it into the new Spend (below), and the new phone signs a `SignerListSet` making the card its rescuer too. (With a co-signer, the phone was never on Save's list, so Save's rules don't change: the new phone just learns who the co-signer is.)
- **Lost card.** Save's signer list still names the phone's Save signer and Save's recovery key, quorum 2. walletd builds a multisigned payment from Save to this phone's Spend, the Save signer and the recovery key (typed in, used, then wiped) each sign it, and their weights meet the quorum. Whoever finds the card holds one signature of three.
- **Phone and card both lost.** The wallet recovery key restores the same wallet key and Save signer, so the new phone holds Spend again, and is again on Save's list. With Save's recovery key it meets the quorum, as above.
- **Lost duress wallet.** Its signer list names the card's duress key, so the card moves it the same way it moves Spend.

Every recovery transaction is built by walletd with a destination it fixes itself (this phone's Spend), carries an expiry, and has its signatures verified before it is submitted. Because the rules are on the ledger, they keep working if Cove's servers, or Cove itself, are gone: any XRP Ledger software that can multisign could perform the same steps.

Card recovery moves everything. Save is linked only once it is found on the network the app is connected to (its address is the same on every network). When the card recovers a lost phone, it signs as the old Spend's signer, and walletd builds every transaction with this phone's Spend as the only destination:

1. the old Spend's XRP first (a multisigned `Payment`), keeping enough for the fees that follow;
2. a trust line on the new Spend for each token (`TrustSet`), signed by the new phone;
3. per tap, up to 40 multisigned transactions: open orders and collectible offers cancelled (`OfferCancel`, `NFTokenCancelOffer`); each token, traded and pool shares alike, paid here in full (`Payment`), and its emptied trust line removed (`TrustSet` with a zero limit); each collectible offered to the new Spend for nothing (`NFTokenCreateOffer` with that destination and a zero price);
4. the new phone accepts exactly those collectible offers (`NFTokenAcceptOffer`). walletd computes their ledger IDs itself from the old account and each offer's sequence, so it can't be handed an offer that asks for money;
5. the XRP those removals freed, in a last tap.

The card stays the old Spend's signer and the old account is never deleted. Anything that couldn't move can still be moved later with the card: for example a token its issuer froze, or a collectible minted non-transferable.

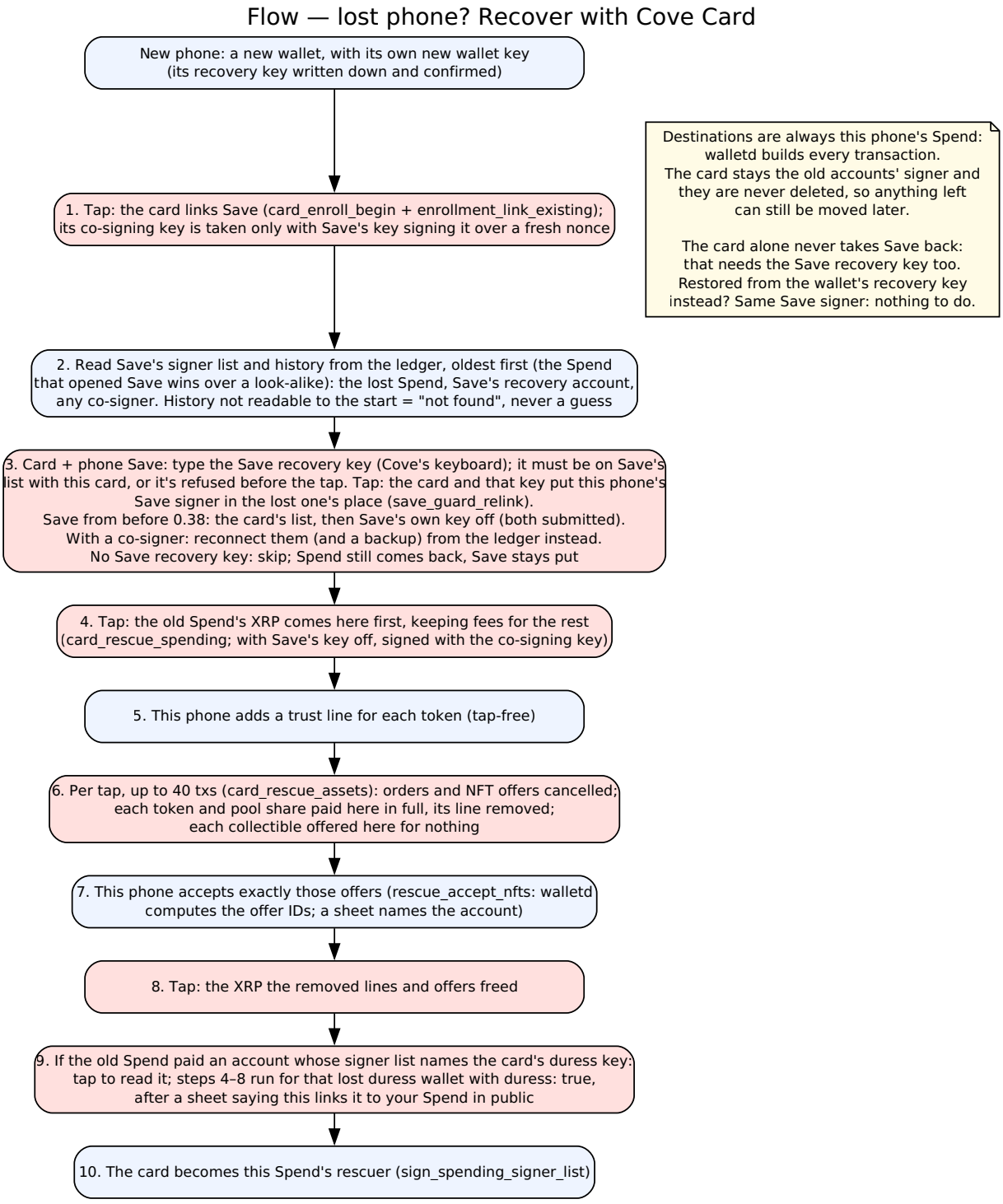


Figure 5. Lost phone? Recover with Cove card. Red steps are card taps.

10. The duress wallet

If someone forces the owner to open Cove, a second PIN opens a small, separate wallet. It has just a Spend account, its own contacts and settings, and nothing that mentions Save, the card or the real wallet.

- **Indistinguishable at rest.** The duress state is sealed in its own slot. When there is no duress wallet, that slot holds random filler of the same size. Every unlock tries both slots the same way, so neither timing nor errors reveal that a duress wallet exists or which PIN was used. Wrong PINs share one counter.
- **Walled off.** walletd refuses every Save, card and duress request while the duress wallet is open.
- **Three days shut.** Once the duress wallet has been opened, the real wallet stays shut on that phone for three days. The real PIN is answered exactly like a wrong one (and counted as one), so a coercer who extracts it learns nothing. The three days must pass in the **network's** time, the close time of the latest validated ledger, and on the phone's own boot clock within one boot, so neither setting the date forward nor restarting the phone shortens them (a restart starts the phone's count again); offline, the hold doesn't end. Each use restarts the three days. Inside the duress wallet nothing gives it away: Save's screens look like a wallet that hasn't set Save up, and its duress settings answer like a wallet without a duress wallet (walletd 0.47). The owner is told this before the duress wallet is created. Recovery on another phone, with the recovery key or the card, still works, because the hold is a rule of this phone's walletd, not of the ledger.
- **Recoverable like Spend.** The duress wallet has the same risk profile as Spend, so the card can rescue it too. The card signs for it with a separate key (`m/44'/144'/0'/0/2`), so the duress account's signer list doesn't name the account that signs for Save and Spend. On a new phone, card recovery finds the duress wallet among the accounts the old Spend paid and moves its funds here. Writing down its own recovery key remains optional.
- **Funding it.** The ledger is public: a payment from Spend to the duress wallet, or back, links the two for anyone who looks the duress wallet up on a block explorer, and from Spend they can find the rest of the wallet. Cove recommends funding it from somewhere else, such as an exchange or another wallet, and shows its address and QR code for that. Funding it from Spend still works, after the owner acknowledges that warning (Cove 0.4.10). Cove doesn't mix or layer the payments to hide the link.

The threat model is coercion by someone holding the phone, not forensic analysis of its storage or of the ledger.

11. Cove Shield and the backend

Cove Shield is a firewall for payments. It is a block list of addresses known to belong to attackers: drained wallets, thieves' receiving addresses, scam destinations. walletd checks it before saving a contact and before signing anything. The check covers every account a transaction names: destinations, issuers, path hops and offer owners. Because the check runs inside walletd, a compromised app interface can't skip it. The user can remove an entry or add their own.

Keeping the list current. The app carries no list of its own. Cove's list comes as a signed feed from Cove's backend, which the app fetches when it first checks in and twice a day after that. Clearing the list in the app removes the user's own entries and downloads Cove's again; walletd keeps Cove's verified list until a newer one arrives. walletd accepts a feed only if:

- it is signed with an offline Cove key pinned in walletd. The backend only stores and serves the bytes, so a compromised backend can withhold an update but never add or remove an entry;
- it is newer than the version already held (no rollback), not expired, and not dated in the future;
- its content is well-formed, with no unknown fields, bad addresses or duplicates.

What the phone is working from. About and Cove Shield list every signed document Cove relies on for the current network (Cove Shield's list and its rapid list, the Trade & Earn tokens and the fee settings): its version and entries, when it was signed and until when it is valid, the key that signed it (its full id), and when walletd last verified it on this phone. A failed check says so in plain words and that Cove keeps using what it verified last. Prices and charts are not signed, and About says so.

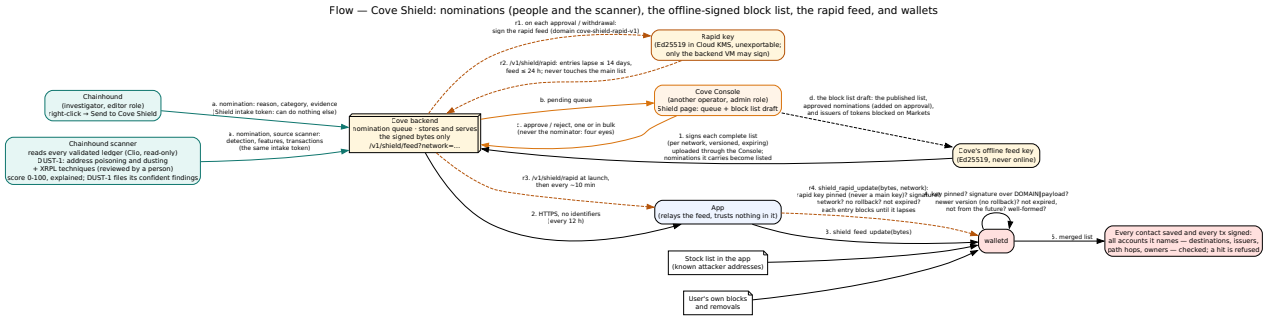


Figure 6. Cove Shield: the block list and its offline-signed feed.

Finding attackers: Chainhound and its scanner. Cove's investigators use Chainhound, a private XRP Ledger explorer, to trace attacks and nominate addresses for the list. Its scanner reads every validated ledger and looks for attack signatures. The first tells apart two attacks built from tiny payments:

- **address poisoning**: the attacker sends a tiny payment from an address made to look like one the victim really pays (the same first and last characters), hoping the victim later copies it from their history;
- **dusting**: tiny payments sprayed at many accounts at once, often with a lure in the memo.

Every finding gets a score from 0 to 100 against Chainhound's rules, with its reasons; known exchanges and the victim's real counterparties are left out. The rules themselves aren't published, so attackers can't tune around them.

Further signatures find attacks built from the XRP Ledger's own features: partial payments that show far more than they deliver (the exchange-crediting exploit), many accounts emptied into one new account (a leaked-key or wallet-bug drain), one regular key or signer added to many accounts that are then emptied (SetRegularKey phishing), tokens copying the code of a known one from another issuer (fake RLUSD), priced NFT sell offers aimed at many accounts, and issuers clawing back or freezing holders' tokens. These never nominate by themselves: an investigator decides each one. Chainhound also keeps the large XRPL incidents on record (among them the Chris Larsen wallet theft of January 2024 and the Bitget hack of September 2026), with the accounts involved checked on the ledger.

The scanner can't block anything. It only nominates, and a person other than the nominator approves every address. The scanner reads only the public ledger and never learns which accounts belong to Cove's users.

Rapid updates. The offline-signed feed changes only a few times a day, so approved addresses also go out on a second, **rapid** feed within minutes. It is signed by a separate online key held in Google Cloud KMS, which can't be exported, and it is bounded:

- every rapid entry lapses after at most 14 days unless the offline feed carries it by then;
- the rapid key can never remove or override an entry on the offline list;
- its signatures can't be mistaken for the offline key's.

The rapid feed is still signed: the online key signs every approval and withdrawal automatically, so a detection waits only for the second person's approval, not for the offline key. An unsigned list would let anyone between Cove and the phone add blocks. walletd enforces the rapid feed from walletd 0.49 (app 0.4.26, which checks for it at launch and every ten minutes or so while in use): each entry blocks like an offline one until it lapses.

Why a backend, and what it keeps. A block list that never changes can't keep up with attackers. Delivering updates, and understanding the attacks aimed at Cove's users, needs a service. Cove has chosen to collect the minimum that service requires:

- **Shield checks happen on the phone.** The backend never learns which addresses a user pays or saves. Fetching the feed sends no identifiers, only the network name.
- **No accounts, names, balances, contacts or keys** are sent to the backend. Nothing from walletd ever is.
- **IP addresses are not kept.** They are used in memory for rate limiting only.
- **Crash reports are off by default.** If the user turns them on, a report holds the error type, a scrubbed message (no addresses, long hex strings or numbers) and Cove's own stack frames. It goes to Cove's own server, not a third party.
- **Notifications are on unless the user turns them off** (Cove 0.4.23), with no third-party push service. The phone registers a random install id, checks in once a day (app and OS version, network) and fetches its messages from Cove's own server while the app is open; iPhones will use Apple's APNs, which Cove calls directly, once it is wired. The same switch covers a **daily device security check**: the phone reports signals such as a jailbreak or root, hooking tools, a debugger, an emulator, no screen lock or a copy not signed by Cove, and the backend answers with a verdict, which is only logged for now. No keys, addresses or balances are sent. An install unseen for 90 days is deleted; turning notifications off, or erasing the wallet, deletes it at once.
- **Market data and announcements** are public data, fetched without identifiers. Announcements are shown as plain text with pictures that must match a hash the notice names, and links only to Cove's own site or the app stores.
- **Reviewed pairs and the fee config are signed offline too** (walletd 0.47), each with its own pinned key and signing domain, so no document passes as another. Once walletd holds a pairs list, buying a token it doesn't mark tradable, or depositing into a pool it doesn't mark earnable, asks the user first; the fee config can move Cove's receiving account and lower the rate, never raise it. Until Cove's offline keys exist, walletd refuses every such document: the app's built-in pairs and pinned account stay in use, and no Cove Shield entries apply (only the user's own blocks). Planned: charts and analysis signed with a separate online key, so a compromised server could at worst distort a chart.

The wallet works with no backend at all: signing, recovery and the stock Shield list need none. The backend's full inventory of what it holds is kept alongside its code (`services/backend/docs/privacy.md`).

12. Threat model

Assumptions. The card's secure element and the phone's secure hardware resist key extraction. The user keeps the written-down keys apart from the phone and the card. The app stores' update channels are not themselves compromised.

Threat	What it reaches	Outcome	Mitigation
Malware running inside the unlocked app	the ability to ask walletd to sign; the Spend key only while it's open	at most the Spend balance, to saved contacts, and only while the Spend key is open (two minutes idle, ten at most)	Spend is a separate account; the Spend key locks and reopens only through the system sheet; Save needs the card and the phone; contacts need the PIN to add; keep Spend small; About's phone rating; jailbreak detection planned
The app shows one thing and asks walletd to sign another (Spend)	one Spend payment	refused unless the user approves walletd's own words on the system sheet	walletd describes the exact bytes it signs; the approval is bound to that one request; the app warns when they don't match what it showed; Android blocks overlays
A compromised signing interface (the Bybit pattern)	what the interface asks to sign	no new payee, no control change, without the user approving walletd's words naming it	destinations limited to own accounts and contacts, each added through the system sheet; control changes refused outside Cove's own flows; typed transactions, fully decoded; a co-signer's walletd checks independently
Phone stolen while unlocked	Spend, until it locks	payments to existing contacts	locks on leaving the app and when idle; sensitive actions need the PIN again
Phone stolen, locked	sealed data only	no loss	Cove PIN and device credential per guess, with growing waits; keys bound to the device
Stored data copied (backup, file dump)	ciphertext	no loss	device-only, hardware-wrapped keys; PIN never stored
Card stolen	the card's keys, behind the PIN	none on Save, even with the PIN (it needs the phone or the Save recovery key too); Spend, if the PIN is guessed (the card is its rescuer)	8–12 digit PIN, card retry limit; move Save with the phone and the Save recovery key
Card stolen with its PIN	one of Save's three signers; Spend's rescuer	the Spend balance; nothing from Save	Save needs a second key (the phone or the recovery key); keep Spend small and the PIN apart from the card
A written-down key found	one of Save's three signers	no loss on its own	Save's quorum needs a second key; set Save up again to replace it
Card stolen with its PIN, and Save's recovery key found	Save's quorum	Save can be emptied	store the recovery key apart from the card, never together
Both written-down keys found together	Save's quorum (the wallet words restore the phone's Save signer)	Save and Spend can be emptied without the card	store them apart; the app says so when each is shown

Threat	What it reaches	Outcome	Mitigation
Part of a written-down key seen	some rows and check digits	fewer guesses for the rest (Section 6)	protect the whole sheet; never split a key across places
The phone shows one thing and asks the card to sign another	one Save transaction	possible loss, within the user's own accounts and contacts	walletd builds transactions; destinations limited to ones approved in the system sheet; **no trusted display on the card (residual)**
Paying an attacker's address	one payment	refused	Cove Shield, enforced in walletd
Address poisoning: a look-alike address planted in the user's history by a tiny payment	one payment, to the look-alike	refused once the address is listed; until then, the user may pay it	the scanner detects address poisoning and nominates it; a person approves; the rapid feed reaches wallets in minutes; full addresses always shown, never shortened
A malicious token or trust line	a token's balance; the line's reserve	no XRP or other tokens lost	lines are only made by the owner; rippling always off (walletd refuses to turn it on); empty lines removable; an issuer can freeze or claw back only its own token
A compromised backend	the Shield feed	offline list: updates withheld, never forged. Rapid feed: addresses blocked for at most 14 days, never one unblocked from the offline list	offline-signed feed with pinned keys and rollback protection; the rapid key is separate, pinned, held in Cloud KMS, every use logged, and its entries lapse
A held-back signed transaction	one transaction	used once, before it expires	mandatory expiry; sequence numbers
Coercion ("wrench attack")	what the user unlocks	loss up to what is unlocked	small Spend; the duress wallet, after which the real PIN reads as wrong on that phone for three days (network time and the phone's boot clock); Save behind the card and the phone; with a co-signer, the Save hold: nothing from Save arrives for days, and the co-signer can cancel it
Cove disappears	availability only	no loss	standard BIP-39 phrases on the standard XRP path; the Cove Key decodes by hand

The hot tier is the largest exposure, by choice. Money for trading, daily spending or instant access is protected by the phone, its secure hardware and the user's credentials. A fully compromised, unlocked phone can lose it. The ledger guarantees it can lose nothing more. Keep in Spend only what you would carry in a physical wallet.

13. Guardian keys: inheritance and assisted recovery without Cove

Cove's own recovery covers the loss of any one thing, and of the phone and card together. Losing two of the user's own keys plus the paper that would replace them, or an owner who has died, needs other people. Today the user arranges that themselves, with **guardian keys**: people they choose, added as co-signers on Save. **Cove is not involved**: it never contacts a guardian, holds a key, or takes part in a signature.

A guardian is a co-signer (Section 4) with a **higher weight** than the owner's own keys. Save's signer list then reads:

Signer	Weight
The owner's card (its co-signing key)	1
Save's recovery key	1
Guardian A (the co-signer)	2
Guardian B (the backup co-signer)	2
Quorum	**3**

So any two of the three parties can move Save, and none alone can:

- **The owner and either guardian**: the card (or the recovery key) and one guardian reach 3. This is everyday use with a co-signer, and recovery when the card is lost.
- **The two guardians together, without the owner**: 2 + 2 reach 3. If the owner has died or can no longer act, the guardians **settle** Save to the address they agree on (Cove 0.4.7), as an estate needs.
- **The owner alone** (card and recovery key, 1 + 1) can't move Save or remove a guardian: a stolen card and a forced owner can't empty it.

Each guardian approves on their own phone with their own Cove wallet, by QR code; their walletd checks every signature already on the request and describes the transaction in its own words. The Save hold (Section 4) still applies: a transaction waits days after a guardian approves, and that guardian can cancel it at any time. Choosing guardians the owner trusts, and telling them what to do, is the owner's job: there is no waiting period or "still alive" check beyond the hold.

Cove Guardian, a proposed service, not implemented. Cove has designed a service that would hold one co-signing key on Save and co-sign recovery or a beneficiary's sweep after waiting periods and notices (`security/guardian.md`, with its own PDF). **It is deliberately not built.** A key Cove held, and rules Cove enforced about when it signs, would make Cove a party to the user's funds: managed custody, not self-custody, with the legal, regulatory and security weight that carries, and a new target for attackers. Until that trade is decided, guardian keys stay with people the owner chooses.

14. What ships today

- **Android and iOS**, on the XRP Ledger: Spend and Save, tokens, trading, liquidity pools, and collectibles as an opt-in developer feature. Android is in the Play Store's internal test track; the iOS app is being prepared for TestFlight.
- **Reviewed markets only.** Trade and Earn offer tokens Cove has reviewed (RLUSD, USDC, SOLO, EURØP, BBRL, XSGD, AUDD and Bitstamp's USD and EUR). On Mainnet a market needs at least $10,000 of trading near its price (its pool plus the order book within 2%); below that it stays listed, dimmed, with buying and deposits off, and selling and withdrawing always on. A **Cove Star** marks the top picks (today RLUSD and USDC): a highlight, not an endorsement.
- **Scan to pay** a shop's or an invoice's QR code, in XRP or a reviewed token, confirmed on the system sheet.
- **Card setup** with the genuineness check, Save's signer list (two of the card, the phone and Save's recovery key), its recovery key and the card as Spend's rescuer. An optional co-signer, always needed, with the Save hold on by default, an optional backup co-signer, and settling a Save without its owner. Optional backup cards.

- **Self-recovery** for each loss in Section 9, including card recovery that moves every asset and a lost duress wallet.
- **A duress wallet** behind a second PIN, which locks the real wallet on that phone for three days (network time and the phone's boot clock), and which the card can recover.
- **Cove Shield** enforced in walletd, with a signed update feed.
- **Notices** on Home for Cove news, features, events and Cove products, with a picture behind the text. Closing one lasts until Cove next unlocks.
- **Notifications and a daily device security check**, on unless turned off, from Cove's own server (Section 11).
- **Signed reviewed pairs and fee config**, checked by walletd, waiting on Cove's offline keys (Section 11).
- **No Cove server in the key path.** The network is needed only to read balances and submit transactions.
- **A paid app, paid on the ledger.** Every swap, buy and sell, and every pool deposit and withdrawal, pays Cove a 0.5% fee on its XRP side, however much is traded, shown before confirming and paid as a separate payment once the trade has gone through, so a failed trade costs nothing extra. Sending, receiving, moving between your own accounts, recovery, trust lines and the duress wallet never pay it, and a token-for-token trade has no XRP side to charge. Cove's account and the rate are pinned in walletd: a compromised app can't redirect the fee or raise it, because walletd signs a fee only after a trade, and only up to 0.5% of that trade. The receiving account can move without an app release, by a config signed offline with a key pinned in walletd (as the Shield feed is), so a compromised backend still can't redirect fees, and no config can raise the rate. Cove is a paid app: a one-off price to download it from the App Store or Google Play, then this fee on trades and pool deposits and withdrawals, and nothing else. No third-party adverts (only Cove's own notices) and no requests for gifts.

Verification status. walletd's behaviour is covered by its own test suite, which drives the card protocol through a simulated card, and by the XRP Ledger codec's checks against the reference implementation's fixtures. The Android build has been run on a real phone with a real Cove Card. The iOS app runs in the simulator; its Secure Enclave path is being tested on a device through TestFlight.

15. Limitations and roadmap

Known limitations.

- The hot tier depends on the phone's integrity while it is unlocked. About's rating checks the secure hardware, the screen lock, Android's attested boot state and the app's signature. The daily security check reports jailbreak, root and hooking signals, but nothing acts on them yet, and platform attestation is still on the roadmap.
- The card has no screen. The phone is its only display.
- Card genuineness is proven with Arculus's card key, which Cove holds and has confirmed with Arculus on two cards; walletd 0.51 doesn't check it yet, and the next walletd release does.
- The EAL6+ certification applies only while the card runs its maker's applet. Cove's own open applet will not carry it.
- Losing the card and Save's recovery key (or the phone, the card and a written-down key) leaves Save unrecoverable unless the owner set up guardian keys (Section 13).

- Hardware trust is delegated to Apple, Google and the card's maker. The current card applet and its PIN handling are closed and not yet independently tested.
- The card's wrong-PIN limit and lockout behaviour are not yet measured.

Roadmap.

1. Device integrity checks before keys are used: acting on the daily security check's verdict, and platform attestation with Apple's App Attest and Google's Play Integrity, feeding About's rating (Insecure stops the wallet).
2. Arculus's card key pinned in walletd: a fresh challenge signed on every card check, and backup cards written only to cards that pass it.
3. A Cove card applet, BSD 3-Clause licensed: a reproducibly built CAP file with on-card attestation. Open and verifiable, not EAL6+ certified.
4. More chains: five networks are planned. The XRP Ledger is first and the only one at launch; Solana, Ethereum, Monero and Bitcoin follow, each its own milestone.
5. Independent audits of walletd, the app and the backend, and an independent assessment of the card, its applet and its PIN handling.
6. A published specification of the Cove Key and the recovery flows.
7. Four-eyes rulesets for the scanner's XRPL technique signatures (their parameters are in code today), and more of them as attacks evolve.
8. A Recovery check screen showing which keys and guardians the user has, and what each loss would mean. Cove Guardian (Section 13) stays a proposal: it would change self-custody to managed custody.
9. **Cove's own cold-storage platform:** an example built on different hardware from today's card, with the open applet above, so every part of the stack can be inspected.

Conclusion

Self-custody doesn't need a new kind of device. The most tested secure hardware there is already sits in people's phones and payment cards. Cove builds on it, and spends its effort on the ways people lose money.

Security, convenience and recoverability need not be traded against each other when each is given to a different thing. The phone's secure hardware makes daily spending fast. A card behind a long PIN keeps savings offline. Two self-checking written keys make recovery possible without anyone's help. Because the rules live on the ledger, in signer lists rather than in an app or a company, no single thing can take the savings, and the owner can always take them back.

Co-signing, a duress wallet and recovery from any single loss are part of the everyday app, not expert settings: secure by default, and simple to use.