MINTID

MintID Holder Manual

What it means to hold a MintID credential: getting one, proving things without revealing yourself, staying valid, and what happens when things go wrong.

VERSION 0.1 · 2026-08-09 · PUBLIC EDITION

DESCRIBES THE DESIGNED PROTOCOL — WALLET AND PROOF ENGINE NOT YET SHIPPED

ECONOMIC FIGURES ARE ILLUSTRATIVE BANDS, PENDING EXTERNAL SIMULATION

> **Status and scope.** Public edition (v0.1). This manual describes the protocol as designed and frozen (`master/SPEC-1` v2.0); no production wallet exists yet and the zero-knowledge proof engine is still a fail-closed stub (Epic F3). Everything here is written in the voice of the designed system, not of a shipping product. Only two numbers are hard protocol constants: the **10-second** presentation and the **30 s heartbeat / 45 s fail-closed** status rule. Economic figures are illustrative bands pending the external simulation.

---

1. The idea in five minutes

Do KYC once; prove it endlessly, without ever revealing who you are. As a **holder** you complete a real identity verification with an approved issuer, exactly once per validity period. What you receive is not an entry in someone's database — it is an **anonymous credential** that lives only on your device, like a document in a sealed envelope with the issuer's seal stamped *through* the envelope: anyone can check the seal, nobody can read the contents.

When a service needs to know something about you, you do not show the credential. Your wallet generates a **zero-knowledge presentation** that answers exactly one question and no more — "is this person over 18?", "is their KYC grade at least A3?", "is the credential still active?" — yes or no, cryptographically guaranteed, with nothing else attached.

The red line the whole network is built on: no personal data, no credential, no serial number and no presentation ever touches the chain. The chain sees only issuer and verifier registries, 32-byte status roots, and token movements (R2). Nothing you do as a holder is recorded anywhere public.

What you actually hold:

- **Your keys** — in your device's secure hardware (or a corporate HSM for fleet operators). Nobody else may custody them: not the network, not the issuer, not any verifier or council (R3).
- **The credential** — the only copy, encrypted on your device. The issuer keeps the *identity link* (who is behind which case file) in its private vault, but not the credential itself.
- **A revocation witness** — a small private proof that your credential is still valid against the issuer's latest published root. Your wallet keeps it current in the background; it is the wallet's only network dependency between presentations.

You may hold multiple credentials from one or more issuers; the protocol never creates a public person-level master identifier (R4).

2. Getting a credential

1. **Complete KYC with an approved issuer.** The issuer — never a validator or verifier — checks your documents, runs liveness checks, and stores the evidence encrypted in its own vault.
2. **Receive the credential offer** over OpenID4VCI, the standard wallet issuance protocol.
3. **Your wallet responds with a holder-bound request** — the credential is cryptographically tied to a secret only your wallet knows.
4. **Receive the anonymous credential plus witness material.** The issuer records your credential's status in its private registry and publishes its next status root to the chain.
5. **Your credential becomes usable** once a chain-finalized issuer root commits to its valid status — normally within seconds.

Assurance grades

GRADE	DEFAULT VALIDITY	MEANING
A1	12 months	Basic identity binding and issuer verification
A2	12 months	Stronger document and liveness verification
A3	6 months	Enhanced due diligence and periodic evidence review
A4	3 months	Highest assurance, stricter issuer controls and audits

Your exact grade is **private** (R6): a verifier can learn "grade ≥ A3" if you agree to prove it, never the grade itself unless you explicitly reveal it. The re-verification cadence is fixed by the protocol, not by the issuer. Renewal is a **new credential** with fresh secrets — publicly unlinkable to the old one, even from the same issuer (R7) — and you may renew with *any* accepted issuer, not just your original one.

3. Proving things: the 10-second presentation

When a registered verifier (a bank, a marketplace) needs something from you:

1. It sends your wallet a **signed challenge**: a one-time nonce, its exact HTTPS origin, the precise claims it requests, and an expiry **10 seconds** away (R14).
2. **Your wallet checks the verifier first** — that its ID, origin and signing key match the on-chain registry and your own local trust decisions (§4). Verifiers must be KYC-approved and registered too (R13); this street runs both ways.
3. Your wallet generates **one anonymous proof** covering exactly the allowed claims, bound to that challenge, that origin, that policy, the issuer's current status root and the chain height — and nothing else.
4. The verifier checks the proof against live chain state and decides.

What the verifier learns: which issuer stands behind you, and the yes/no answers it asked for (`age_over_18: true`).

What the verifier never learns: your name, your attributes, your exact grade, your credential's expiry date, any credential identifier — and it cannot recognise you the next time. Two presentations, even to the same verifier, are unlinkable; presentations to different verifiers can never be correlated (that is a tested acceptance criterion, not a hope).

What survives afterwards: nothing on-chain, ever (R18). The verifier keeps a minimal five-field decision record with no personal data in it — the format structurally cannot carry your attributes.

The 10-second window makes every presentation perishable: nothing can be replayed, stockpiled, or reused at another site. One session can prove several predicates at once under an explicit policy (R22).

4. Trusting verifiers: your local pin store

Your wallet keeps a **local, private** list of verifiers you have approved: `(verifier ID, exact HTTPS origin, key fingerprint)` (R23). Wildcards do not exist (R24); trust-by-name does not exist. If a verifier rotates its keys, changes origin, or its KYC status changes, your wallet asks you again before presenting. This approval list never leaves your device — it is invisible to verifiers, validators and the chain itself.

5. Staying valid: the heartbeat and fail-closed

Your issuer publishes a fresh status root to the chain every **30 seconds** (R9). Your wallet quietly refreshes your witness against each new root. Verifiers reject any proof built on a root older than **45 seconds** — always, with no exceptions: staleness *fails closed*.

What this means for you:

- A revocation (yours or the issuer's) takes global effect within at most 45 seconds.
- If your issuer stops publishing — outage, suspension, shutdown — your credential stops verifying everywhere within 45 seconds. Nothing on your device changes; proofs simply stop matching a fresh root.
- The remedy the protocol gives you is a **new credential from any accepted issuer** (§2). There is no witness escrow or issuer failover: the fail-closed rule is deliberately stronger than any promise an issuer could make.

6. Self-revocation: the kill switch you always hold

You can revoke your own credential at any time, **without the issuer's cooperation**, and the revocation transaction reveals **neither who you are nor which credential died** (R11). The network only sees that *some* valid credential of that issuer asked to stop existing: your wallet derives a one-time nullifier and proves in zero knowledge that it comes from a valid credential — without saying which.

Use it the moment a device is lost or a key might be compromised: the credential dies everywhere within 45 seconds. For agent fleets (SPEC-2), each agent credential can be killed individually without revealing the principal or any sibling, and a principal-wide cascade must not leak even the fleet's size.

> Self-revocation is classified as a high-risk cryptographic component with its own circuit specification, test vectors and independent audit. **It will not be enabled on mainnet until those audit gates pass.**

7. Lost keys and recovery

There is no social recovery, no seed phrase, no cloud backup. The designed recovery path is exactly one (R12, §12.2):

1. Wallet key lost or compromised → **self-revoke** if you still can (§6); the lost device's credential dies everywhere in 45 seconds.
2. Request **fresh KYC re-verification** with an accepted issuer. The issuer may identify and revoke your prior credential in its private records.
3. Receive a **brand-new credential** with new secrets — publicly unlinkable to the old one.

This is self-custody: the device and its PIN are your responsibility, like a bank card. The difference is that losing it never exposes your data — because the data is not in it. For escrowed funds in the agent layer, spend authority is deliberately separate from the credential secret: a leaked credential secret can never move value out; the worst case is delay, not loss.

8. What it costs (and what it never costs)

Verification is free, by construction, forever (R15/R18): presentations are off-chain and no usage tax exists. What has a price:

ITEM	WHO PAYS	STATUS
KYC itself	You/your issuer relationship — priced off-protocol by the issuer	market ~$0.80–1.85 per automated check
Agent mint fee (agent layer)	Principal → issuer, per agent onboarded	illustrative band $1–5 (base $2)
Agent lease (agent layer)	Principal → issuer, recurring per agent-year	illustrative band $6–24 (base $12)
Consensual disclosure	**You get paid** — the disclosure bid goes to the holder	illustrative band $5–50 (base $15)
Self-revocation tx fee	Via a privacy relay; relayer-funded fees are a could-have	open

All bands are draft analyst figures — none is ratified; the flow-A parameters are not even implemented yet. Your issuer privately enforces an active-agent cap per holder (anti-sybil, R-A7); no numeric cap is published.

9. When can your identity ever be revealed?

Only through due process — never by payment, never by the issuer's own choice, never silently:

- **L0 — money first, no identity.** Commercial disputes settle against locked escrow, revealing nobody. The claimant posts a bond, slashed if the claim is frivolous.
- **L1 — a minimal, PII-free fraud signal** if escalation is warranted.
- **L2 — identity disclosure, due process only.** A neutral, bonded, sortition-selected arbiter — whom neither party picks — must certify that both parties rejected settlement *and* that a court is genuinely warranted. Only then does **the issuer** (the only party holding the identity link) release it, under that certified finding, with the council supervising and auditing the release. Disclosure is never purchasable (R-A22), and the minimum-tier rule applies: pseudonym before contact attribute before full identity (R-A20).

On the separate **consensual** rail, nothing is disclosed without your explicit consent policy; a non-disclosing path to the core transaction must always exist (R-A23), and full identity is never delegable to an agent — it requires you, live (R-A21.1).

Annex – Your privacy guarantees at a glance

#	GUARANTEE
R2	No documents, biometrics, names, serials or presentation logs ever on-chain.
R3	Nobody custodies your keys – not the network, issuer, verifier, validator, council or custodian.
R4	Multiple credentials, no public person-level master identifier.
R6	Your exact grade is private; only threshold proofs leave your wallet.
R7	Renewals are new, publicly unlinkable credentials.
R8	Your status is a private witness – no per-user state on the chain.
R11	Self-revocation reveals neither your identity nor your credential.
R12	Recovery produces a new, publicly unlinkable credential.
R14	Presentations live 10 seconds and bind totally – nothing is replayable.
R18	Presentations never enter the chain; verifiers keep only a minimal decision record.
R21/R22	Predicates, not attributes – one session, exactly the questions allowed.
R23	Your verifier approvals are local-only, invisible to everyone.
R25	Even council actions publish only a reason code and an evidence commitment.
R-A2/R-A3	Your agents are mutually unlinkable; the only link lives in the issuer's escrow.

Manual v0.1 (public edition) – 2026-08-09. Sources: `master/SPEC-1` *v2.0 (§4, §10, §12),* `master/SPEC-2` *(agent layer, R-A series),* `master/SPEC-3` *(disclosure and remediation),* `docs/economics/` *draft report. No wallet product exists yet; the proof engine is a fail-closed stub until Epic F3 ships with its audits.*