



MintID

ROLE MANUAL

OCTOBER 2026

MINTID · ROLE MANUAL

MintID Holder Manual

Verify once, prove what a service needs, and keep who you are to yourself —
a plain guide for people and for anyone who deploys AI agents.

PUBLISHED BY

MintID Foundation

Netherlands Stichting, Amsterdam · related parties disclosed

VERSION

0.2 · October 2026

AUDIENCE

People and agent operators

NETWORK

Single-node production

VERIFICATION

Free by design

Contents

01	MintID in five minutes	3
02	Getting your credential	4
03	Proving things in ten seconds	6
04	Staying valid: the 30-second heartbeat	7
05	Your kill switch: self-revocation	7
06	Lost device, lost keys: recovery	8
07	Credentials for your AI agents	8
08	Funded recourse: escrow you keep	9
09	When your identity can be revealed	10
10	What it costs	11
11	When something goes wrong	12
12	Where MintID stands today	12
A	Your guarantees at a glance	13
B	Plain-language glossary	14



Status and scope. MintID runs in production on a single-node chain, which is never a place to test; the distributed mainnet has not launched. Anyone building holder software tests on **the public testnet**, with its KYC sandbox: the same software and parameters, a KYC session whose outcome the caller chooses, and nothing issued there carries weight; whether it is live is stated in one place, the developer documentation's page *The public testnet and its KYC sandbox* ([docs/sdk/17-sandbox-and-testnet.md](#) in the main repository). MintID ships no wallet: the holder library runs in the infrastructure of the organisation that holds the credential (on the ZadQ rail, the seller side), and a wallet somebody else builds can present over the same transports. This manual describes the protocol as designed, in the present tense for readability. The proof engine that holders depend on is built and in production (chapter 12). Timelines are targets, not promises. This manual is not legal advice and not an offer of any crypto-asset: nothing is for sale.

01

MintID in five minutes

Verify your identity once with a regulated issuer, then prove what a service needs to know, as often as it needs to, without saying who you are.

The idea. Proving something simple about yourself — that you are an adult, or that a bank has checked your identity — usually means handing over a copy of your identity document, which the other side keeps. MintID separates the claim from the person. You complete a real identity check (KYC) once, with an **accepted issuer**: a regulated organisation that the network has admitted and that has posted a financial bond. You get back an **anonymous credential** that lives only in your own wallet. Think of a sealed envelope with the issuer's seal stamped *through* it: anyone can check that the seal is genuine, but nobody can read what is inside.

When a service needs to know something about you, you never show the credential itself. Your wallet — the plain word this manual uses for your holder software, the headless library MintID ships or a wallet built on the same transports — produces a **presentation**: a zero-knowledge proof that answers exactly the questions asked — “*is this person over 18?*”, “*has this person been verified at grade A3 or higher?*”, “*is the credential still active?*” — with a yes or a no, and nothing else. The service learns those answers and which accepted issuer vouched for you. It does not learn your name, and it cannot recognise you the next time you visit.

0

personal records, credentials
or proofs ever written
on-chain

10 s

lifetime of each proof you
give; it can be used only once

30 s

how often your issuer
confirms status; stale after
45 s

Free

basic verification, by design;
no usage tax

Why there is a blockchain at all. Someone has to keep a public, tamper-proof record of which issuers and services are approved and whether each issuer's credentials are still in good standing. MintID's chain does that and nothing more: registries, a small status commitment from each issuer every 30 seconds, the one-time codes that switch credentials off (chapter 5), and the network's own governance and economics. It never holds anything about you, and it grows with the number of organisations, never with the number of people.

What you hold.

- **Your keys**, in your device's secure hardware (fleet operators may use their own hardware security modules).



- **Your credential**, only on your device and bound to a secret that never leaves it. Your issuer keeps your KYC evidence in its own encrypted records, but it cannot use your credential.
- **A status witness**, a small private proof that the credential is still valid, refreshed in the background.
- **Your trusted-services list**, the services you have approved, kept privately on your device (chapter 3).

You may hold several credentials from one or more issuers; nothing public ties them to one person.

Nothing about you goes on-chain: no documents, biometrics, names, credential numbers, proofs or activity.

Your keys stay yours: no network, issuer, service, validator or council holds them.

Each proof answers only what was asked, lasts ten seconds, works once and cannot be used to recognise you.

You can switch your credential off yourself, without anyone’s help or revealing who you are (once it is built).

Your identity leaves only with your own authorisation — your consent, or a bounded mandate you gave — or through due process, never because someone paid.

Who you will meet.

PARTY	WHAT THEY DO FOR YOU	WHAT THEY CAN NEVER DO
Accepted issuer	Checks your identity, issues your credential, confirms its status	Put your data on-chain or see your presentations
Service (verifier)	Asks for the minimum it needs and checks your proof	Ask for unapproved claims, reuse a proof, log you on-chain
Validator	Runs the network that keeps the public records honest	See any KYC document or judge who you are
Compliance council	Admits, supervises and sanctions issuers and services	Approve or reject a person, or change the software

02

Getting your credential

One identity check with an accepted issuer gives you a private credential that you, and only you, control.

STEP 1 Choose an issuer Any accepted issuer: admitted on-chain (by the compliance council once it is seated) and bonded your choice	STEP 2 Verify once Documents and liveness checks; the evidence stays encrypted with the issuer off-chain	STEP 3 Receive it Bound to a secret only your wallet knows; the only copy is on your device your device	STEP 4 Status confirmed Your issuer's next status commitment is finalised on-chain; nothing names you next heartbeat	STEP 5 Ready to use Prove things to any registered service, as often as you need until it expires
--	---	--	---	--

How it works. Only the issuer checks your identity; the network, validators and services never see your documents. Your wallet receives the credential over the same open standards (OpenID4VCI and OpenID4VP) that the European Digital Identity Wallet uses. Because the credential is tied to a secret only your wallet knows, a copy is useless to



anyone else. It becomes usable once your issuer's next 30-second status commitment is confirmed on-chain. No transaction records that a credential was issued, and getting or using one involves no transaction of your own.

Assurance grades

Every credential carries a private **assurance grade** from A1 to A4. The grade describes how *strongly* the issuer verified you, and each grade is anchored on the European eIDAS assurance levels, cross-checked against the international standard ISO/IEC 29115.

GRADE	DEFAULT VALIDITY	WHAT IT MEANS	EUROPEAN ANCHOR (EIDAS)
A1	12 months	Basic identity binding and issuer verification	Low
A2	12 months	Stronger document and liveness verification	Substantial
A3	6 months	Enhanced due diligence and periodic evidence review	Substantial, plus enhanced due diligence
A4	3 months	Highest assurance, with stricter issuer controls and audits	High

The default validity counts from the day your verification was approved, and your credential expires exactly when that verification does — there is no grace period and no extension, and the way back is a new verification. It is also an upper limit: your issuer may ask you to re-verify sooner when its risk rules require it. Your exact grade is **private**. A service can ask you to prove “*grade A3 or higher*” and learns only yes or no, never the grade itself unless you choose to reveal it.

Freshness is separate from grade. How *recently* you were verified is a second, separate measure: a freshness band that says whether you were verified within the last 365, 180, 90 or 30 days (F1, F2, F3 or F4). A service can ask for a minimum grade and a minimum freshness at the same time. It receives a yes or a no, never a date, because a date could be used to recognise you.

Renewing and upgrading. Renewal always produces a brand-new credential with fresh secrets, from your original issuer or any other accepted issuer, and it cannot be publicly linked to the old one, even when the same issuer issues both. A higher grade likewise means a new credential after a stronger check: nobody can quietly edit a credential that already exists.

03

Proving things in ten seconds

A presentation answers a service’s questions with a proof that lasts ten seconds, works once and reveals nothing else.

STEP 1 A service asks A signed, one-time request naming the service, its web address and its exact questions expires in 10 s	STEP 2 Your wallet checks it Registered? Approved? The same service you trusted before? on your device	STEP 3 You decide A new service, or any change to a known one: your wallet asks you first your consent	STEP 4 One proof A single zero-knowledge proof answers all the questions, tied to this request yes or no only	STEP 5 Checked once The service re-checks the chain, accepts once and keeps a minimal record never on-chain
--	--	--	---	---

Services are accountable too. A service that asks you for proof must itself pass KYC with an accepted issuer, register its exact web addresses and signing keys on-chain, prove it controls those addresses and be approved by the compliance council. Its addresses are re-checked periodically, and wildcards are not allowed.

One proof, bound to one moment. Your wallet answers every question with a single proof, tied to this service, this web address, this request, these questions and the current state of the chain. Before accepting it, the service re-checks that your issuer is active and its status current, then marks the request as used, so the same proof can never be accepted twice.

WHAT THE SERVICE LEARNS	WHAT THE SERVICE NEVER LEARNS
The yes/no answers to the questions it was allowed to ask	Your name, your document, your date of birth or any other detail about you
Which accepted issuer vouched for you	Your exact grade, your credential’s expiry date or any credential number
That the proof is fresh and was made for this request	Whether it has seen you before; two proofs, even to the same service, cannot be linked

Why ten seconds. A proof that expires almost at once is worthless to anyone who steals, stores or forwards it. The limit is fixed in the protocol; no service or software library can extend it. Afterwards, nothing remains on the chain, and the service keeps only a minimal record of its own decision, with no personal data about you.

Deciding whom to trust

Your wallet keeps a **private list of the services you have approved**, each tied to the service’s registered identity, exact web address and signing key. Trust is never granted on a name alone, and the list never leaves your device.

If anything about an approved service changes (a new key, a different address, a change in its KYC standing, or closer supervision by the council), your wallet asks you again before answering. Treat that prompt like a bank’s warning about an unusual payment: approve it only if you expected the change. A look-alike site on another address simply does not match, and your wallet sends nothing.

04

Staying valid: the 30-second heartbeat

Your issuer confirms the status of its credentials every 30 seconds, and a confirmation older than 45 seconds is never accepted.

Every accepted issuer publishes a small, signed **status commitment** to the chain at least every 30 seconds. It covers all of that issuer's credentials at once and says nothing about any single person, and your wallet refreshes your status witness against each one. Services use the latest confirmed commitment and **reject anything older than 45 seconds**, without exception. Staleness *fails closed*: when in doubt, the answer is no.

What this means for you.

- **Issuer revocation is fast.** When your issuer revokes a credential, the change is in its next commitment, within 30 seconds, and every service refuses the credential in under a minute. In an emergency the issuer can use a separate emergency key to publish an update immediately. These are the protocol's reference timings; a deployment can govern a longer window, and its operator publishes it.
- **A silent issuer takes its credentials offline.** If your issuer stops publishing, for example because of an outage, a suspension or a shutdown, its credentials stop verifying everywhere within 45 seconds. Nothing changes on your device; your proofs simply no longer match a current commitment.
- **Your remedy is a new credential from any accepted issuer** (chapter 2). There is no automatic switch to another issuer: the fail-closed rule protects you more strongly than any promise an issuer could make.

Issuers are held to this: each posts a bond that the council can confiscate ("slash") for proven misconduct, including deliberately failing to publish status.

05

Your kill switch: self-revocation

You can switch off your own credential at any time, without your issuer's help and without revealing who you are or which credential it was.

A plain "revoke my credential" message would have to name the credential and expose you. Instead, your wallet derives a **one-time code** (a *nullifier*) from a secret in your credential and proves in zero knowledge that it comes from a valid credential of that issuer, without saying which. A **privacy relay** carries both to the chain and pays for the transaction, so the submission does not come from you and you need no account. The chain checks the proof and records the code. Your issuer — the only party that can tell which of its credentials it was — removes it at its next status update, and the chain stops accepting that issuer's updates if it does not; either way your credential then fails every "not revoked" check.

When to use it. The moment a device is lost or stolen, or whenever you suspect someone else may have access to your wallet. Each agent credential can be switched off the same way, individually (chapter 7).

In production on the single-node production chain. As the one zero-knowledge proof the chain itself checks, self-revocation has its own specification, test suite and adversarial testing. It has been in production on the single-node production chain since 24 September 2026: you can revoke your own credential through the relay; your issuer must remove it within a short grace the chain enforces, and every service refuses it in under two minutes. It runs on the



public testnet as it does there, and is switched off at the distributed public network's genesis, where switching it on is a governance decision. Its independent cryptographic audit is a scheduled milestone whose findings will be published. Where it is not available to you, use the route in chapter 6.

06

Lost device, lost keys: recovery

There is exactly one way back from a lost or compromised wallet: a fresh identity check and a brand-new credential.

Nobody holds a spare key: there is **no social recovery** and **no custodial back door**.

1. **Switch off the old credential** if you can (chapter 5), so that it stops working everywhere.
2. **Ask an accepted issuer to verify you again.** It can find your previous credential in its private records and revoke it.
3. **Receive a brand-new credential** with new secrets, usable after the next status commitment and publicly unlinkable to the old one.

This is self-custody: your device and its unlock code are your responsibility, like a bank card and its PIN. Your KYC documents were never on the device, so losing it does not expose them.

For agent operators. A compromised *agent* is switched off and replaced. If your *own* credential is compromised, you re-verify, and your issuer can revoke and re-mint your agents; escrowed money follows its own rules (chapter 8).

07

Credentials for your AI agents

Your agents can prove that a verified, accountable person stands behind them, without saying who.

Minting, not registering. As a verified person you become a **principal**. You prove to your issuer, in zero knowledge, that you hold a valid credential of at least the grade the agent needs, and the issuer **mints** a separate agent credential with its own secrets. The only link between agent and principal stays in the issuer's private records: no public agent registry, no per-agent record on the chain, no public reputation score.

What an agent can prove. In the flow of chapter 3, an agent proves statements such as *“backed by a verified person”*, *“backed at grade A2 or higher”*, *“allowed to perform this action”*, *“allowed to spend at least this amount”* and *“credential active”*, to services, to other agents and alongside x402 pay-per-request payments.

Scope, fixed at minting. You set the agent's **scope** (what it may do, with whom, in which categories and places, when, and up to what value) within your issuer's policy and your own grade. It is signed into the credential, so the agent cannot raise its own authority and a hijacked agent can do no more than its scope allows. The issuer can also state how the agent's keys are held and who operates it, as a signal for services.

Bounded agents. Instead of assembling a scope yourself, you can pick a named **bundle** from your issuer's catalogue, such as paying named sellers for digital goods, buying from named suppliers or reading paid data. A bundle fixes what the agent may do and caps everything else: sellers, categories, regions, value per transaction and lease. You choose only inside those bounds. A bounded agent is always private: it proves what any agent proves and discloses nothing about you.



Your agents cannot be linked. Two agents of the same person cannot be linked on the chain, in their proofs or across services. If you agree, an agent can show a service a **pairwise pseudonym**: a stable name that lets that one service recognise it over time, and that no other service can connect.

Fake armies are expensive. Minting requires a verified person, a per-person cap on active agents, a recent enough identity check, and a mint fee plus a recurring lease per agent, all enforced privately by your issuer.

Three ways to stop an agent.

LEVER	WHO USES IT	EFFECT
Switch off one agent	You	Every service refuses that agent in under two minutes; neither you nor your other agents are revealed
Issuer revocation	Your issuer	In the issuer's next commitment and refused everywhere in under a minute, or at once with the issuer's emergency key
Switch off all of them	You, under your issuer's policy	Revoking your own credential can revoke every agent derived from it, without revealing you or how many agents you run; to keep that number hidden, the agents are switched off over several commitments, a few minutes

08

Funded recourse: escrow you keep

An agent can also prove that real money stands behind it, held in your own wallet under a lock that the agent itself can never open.

Some counterparties want to know that harm can be paid for. An agent can prove that **at least a stated amount** of regulated stablecoins (**USDC or EURC**) stands behind it, held in *your own* wallet under a smart-contract lock on the chain where the stablecoin lives; MintID never runs that contract. MintID never takes custody of your money: no fiat custodian, no third-party deposit, no privacy coins. Lower funding levels are proved from your balance; higher levels need the audited lock.

The agent cannot move your money. The secret an agent uses for proofs has no power over the lock. Withdrawing at your discretion needs a separate release key that exists only on your side, used directly or through your own release policy, which refuses anything outside your rules. A leaked agent secret cannot drain the escrow.

Exactly three ways out of the lock.

1. **Rebalancing within the same agent**, which never moves value to anyone else.
2. **A payout to a claimant** after a dispute verdict. No party's key is involved; the verdict alone decides.
3. **A return to you**, under your release key, or automatically when the lock ends.

There is no fourth way. A session can briefly reserve part of the funds, so the same money never backs two sessions at once. If you lose your own key, the worst case is delay, not loss: the funds return to your wallet when the lock ends. A contested claim **freezes** the reserved funds until decided; neither withdrawing nor switching the agent off can get ahead of the verdict.

Staged. Escrow and disputes open in Stage 3 (chapter 12), once the escrow contract and its proofs are built; their audits are published milestones.



09

When your identity can be revealed

Your identity leaves the system only in two ways, both tightly controlled: your own authorisation (your consent, or a bounded mandate you gave), or due process.

For your own credential. The only link between you and your credential sits in your issuer's encrypted records, under the rules that bind every regulated identity provider. Outside the protocol, the issuer answers only to the legal obligations that already bind regulated identity providers; inside it, no service, validator or buyer can obtain that link.

For your agents, there are two strictly separate paths, and one can never turn into the other.

With your consent: the commercial path

A business may legitimately want to *know* who it deals with, for example to open an account. It can offer to pay for one defined item of information about the person behind an agent, asking only for the lowest step that serves its purpose.

STEP	WHAT IS REVEALED	CAN YOU PRE-APPROVE IT?
Pseudonym	A stable name for that one business; no personal data	Yes, within limits you set
One contact detail	A single attribute, such as an email address	Only for businesses on a named list, with caps and an expiry date
Full identity	Your verified KYC identity	Never: you must say yes yourself, at that moment

- **Off unless you switch it on.** Commercial disclosure is a setting fixed at minting, private by default. Even when on, **refusal is the default:** a timeout, an error or a missing rule means nothing is disclosed and nothing is paid.
- **Your basic service is never the price.** A business may offer extras or an ongoing relationship in return, but must always leave a way to complete the core transaction without disclosing anything.
- **You are paid, but payment is never the reason.** The offer goes to you, minus a protocol fee and any issuer handling fee, yet **no payment is ever a lawful basis for disclosing personal data:** only your own authorisation is.
- **Withdrawing is free.** Narrowing or withdrawing consent costs nothing; widening it requires a new, paid minting.
- **Identification, not surveillance.** A disclosure identifies one agent at one moment and gives no right to recognise, contact or track you later: to know again, a business must ask and pay again, and nothing carries over a renewal.
- **Warned before linking.** Before you disclose personal details to a second business, you are warned that they could then connect your agents, and the pseudonym is offered first.

Under your mandate: identification the law requires

Some businesses must by law record who their client is: a hotel registering its guests, a car-rental desk, a carrier recording its passengers. When your agent books or contracts for you with one of them, you can give the agent a **mandate** in advance, so you are not asked every time.

- **The law fixes the fields, not the business.** Each kind of obligation is a governed *obligation class* that names the rule, the act and the exact fields (for example your names and a document). The business cannot ask for more, and your age is never a field: it is only ever proved.



- **You name who, how often and until when.** The mandate names the businesses (or the kind of obligation), a cap per period and an expiry no later than the agent credential. The first time towards each business, you confirm live.
- **Your issuer delivers, your agent never sees it.** The issuer reads the fields from its identity provider at that moment, sends them signed straight to the business and keeps nothing; the agent only passes the request along and cannot consent on your behalf.
- **Receipted and revocable.** Every delivery is receipted, and you can revoke the mandate at any time, for free.

Built, switched off. This path, like the commercial path above, is built in the issuer software and ships switched off: nothing is disclosed until counsel has settled, country by country, what makes such a mandate valid and which obligations it covers.

Through due process: the dispute path

A wrongdoer would always refuse consent, so disputes cannot depend on it. This path follows a strict **money-before-identity** order:

LEVEL	WHAT HAPPENS	IDENTITY REVEALED?
L0 — Money	The claimant is paid from the agent's escrow	No
L1 — Signal	A minimal fraud signal: a real verified person, a grade, a jurisdiction, a count of earlier upheld disputes	No personal data
L2 — Due process	Full identity, released only through the legal route	Yes, as the last resort

A claim needs a bond and a zero-knowledge **proof of harm** tied to the disputed session, and can never exceed the coverage the agent proved. If you contest it, a **bonded arbiter**, drawn at random weighted by stake so that neither side picks the judge, decides without learning who you are; the loser pays the arbiter, and the decision executes against the escrow. Identity is revealed only if both sides reject that decision **and** the arbiter certifies that a court is genuinely needed. Even then, **the arbiter decides, the issuer releases, and the council supervises afterwards**, never as a precondition. The governance rules for this step are finalised with legal counsel before Stage 3. On either path, the chain sees only a sealed receipt that a disclosure happened, never its content.

10

What it costs

Checking a proof is free by design; the costs that do exist are paid to the organisations that do real work.

ITEM	WHO PAYS WHOM	PRICE
Proving things to a service	Nobody	Free by design. Proofs never touch the chain, and no usage tax exists or is planned
Identity check and renewals	You, to the issuer you choose	Set by the issuer, outside the protocol

ITEM	WHO PAYS WHOM	PRICE
Agent mint and lease fees	You, to your issuer, per agent	Priced per agent; the schedule is set before genesis by the independent economic simulation, and part of each fee goes to the protocol
Consented disclosure	A business, to you	Offered by the business; you may refuse; a protocol fee and any issuer handling fee are deducted
Withdrawing consent	No one	Free
Self-revocation	Sent through a privacy relay	The relay pays the network fee; you need no account and hold no tokens
Escrow	Your own stablecoins	Stays yours; it leaves the lock only by one of the three exits in chapter 8

No figure in this manual is an offer term, and MintID states no token price anywhere.

11

When something goes wrong

Most problems have a single, simple remedy, and the protocol is designed so that the safe outcome is the default.

SITUATION	WHAT HAPPENS	WHAT YOU DO
Phone lost or stolen, or keys exposed	The credential stays locked in your wallet, but act fast	Switch it off (once enabled), or re-verify and have it revoked (chapter 6)
Your issuer goes silent or is suspended	Its credentials stop verifying within 45 seconds	Get a new credential from any accepted issuer
Your wallet flags a changed service	Its key, address or standing is different	Approve only if you expected it; otherwise walk away
An agent misbehaves or is manipulated	It cannot act beyond its signed scope	Switch that agent off, or all of them at once
A business demands your identity for basic service	That breaks MintID's rules for businesses	Refuse: a non-disclosing path must exist
Someone claims your agent caused harm	Money first, identity last (chapter 9)	Settle from escrow, or contest before an arbiter

12

Where MintID stands today

MintID is being built in public stages, and each stage opens only when its published conditions are met.

Today. The chain, the proof-checking service and a first issuer service run in production on the single-node production chain (since 20 September 2026), operated by the CONECTIA group — the MintID Foundation is being



incorporated and stays dormant until the public-offer route is chosen — and the **proof core** that makes presentations possible is built and in production. Self-revocation has been in production on the single-node production chain since 24 September 2026, runs on the public testnet as it does there, and is switched off at the distributed public network’s genesis, where switching it on is a governance decision; its independent audit is a scheduled milestone, as is the audit of the proof core’s implementation, whose findings will be published. MintID ships no wallet; the holder library is published under a permissive open-source licence, and a wallet somebody else builds can present over the same transports.

STAGE	TARGET WINDOW	WHAT IT MEANS FOR HOLDERS
0 — Single-node production (now)	now → 2027	Specifications frozen, a single-node chain in production, proof core in production; a holder library, no consumer wallet
1 — Public offer	2027	A regulated step under the EU MiCA rules; it changes nothing about how credentials work
2 — Identity-only mainnet	H2 2027 → 2028	Credentials, ten-second presentations, free verification, and agent credentials proving human backing, grade and scope
3 — Financial recourse	2028 → 2029	Escrow, disputes and consented disclosure for agents
4 — Maturity	2030 → 2031	More issuers and a growing agent population

Windows are targets. Independent security, cryptography and economic audits, a public adversarial testnet and a funded bug bounty are committed milestones, each published with its findings; if a gate condition is unmet, the plan waits.

APPENDIX A

Your guarantees at a glance

GUARANTEE	WHAT IT MEANS FOR YOU
Nothing personal on-chain	No documents, biometrics, names, credential numbers, proofs or activity, ever
Your keys, your credential	No one else holds your keys; no public identifier ties your credentials together
Private grade, dateless freshness	Services learn “grade X or higher” and a freshness band, never the exact grade or a date
Unlinkable proofs and renewals	Single-use ten-second proofs; no two proofs, and no old and new credential, can be linked
Only what was asked	Permitted questions only, answered yes or no; your trusted-services list stays on your device
Fail-closed status	A status confirmation older than 45 seconds is never accepted
Your own kill switch	Self-revocation reveals neither you nor your credential
No back doors	Recovery only by a fresh identity check; no social recovery, no custodial key
Unlinkable agents, your money	Agents cannot be linked to each other or to you, and cannot move their escrow
Consent or due process only	Payment never justifies disclosure; refusal is the default; disputes settle with money first

APPENDIX B

Plain-language glossary

Agent credential — a credential an issuer mints for one of your AI agents, carrying proof of human backing and a signed scope. **Assurance grade** — A1 to A4: how strongly you were verified; kept private. **Freshness band** — F1 to F4: how recently you were verified, shown as a band, never a date. **Nullifier** — the one-time code that switches a credential off without identifying it. **Pairwise pseudonym** — an agent's stable name for one service, unlinkable across services. **Presentation** — a ten-second, single-use zero-knowledge proof answering a service's questions. **Privacy relay** — a forwarding service that submits your self-revocation, so that it does not come from you directly. **Scope** — what an agent is allowed to do, signed into its credential when it is minted. **Status commitment** — the small signed summary each issuer publishes every 30 seconds; it names no one. **Status witness** — your private proof that your credential is still valid against that commitment. **Zero-knowledge proof** — a proof that a statement is true which reveals nothing beyond that fact.

MintID Holder Manual v0.2, September 2026 · prepared by the MintID Foundation, a Netherlands Stichting · non-normative; the specification corpus governs · not legal advice · not a MiCA crypto-asset white paper and not an offer of any crypto-asset.

