MINTID PROTOCOL WHITEPAPER

A Sovereign Layer-1 for Privacy-Preserving, KYC-Backed Identity Credentials

Verified once, proven anywhere, revealed nowhere: how MintID lets regulated issuers vouch for people and the AI agents they deploy without putting a single identity on a public ledger.

PUBLISHED BY

MintID project

A CONECTIA-group initiative · related parties disclosed in §11.4

Contents

ABSTRACT

Abstract

Reading guide and legal posture. This is the *protocol* whitepaper of MintID. It explains what the system does and why it is designed the way it is: the architecture, the standards choices, the token economics as ratified so far, the governance model and the launch plan. It deliberately stays above the level of wire formats and implementation detail; the frozen specification corpus that defines them is available on request to prospective issuers, verifiers, validators, auditors and counsel, and where this paper and the corpus differ, the corpus governs. This paper is **not** the crypto-asset white paper required by Regulation (EU) 2023/1114 (MiCA) Title II. That document, when it exists, will be the only document so labelled; until it is notified to the competent authority, **nothing is for sale, no price exists, and no figure in this paper is an offer term.** Every economic value below is a *ratified prior* — an argued working point that the independent pre-genesis economic simulation validates or argues against — unless it is marked as frozen. Items marked *(counsel)* are subject to qualified legal review.

Identity on the internet is broken in two opposite directions at once. Where it works, it works by **surveillance**: every login, age check and payment leaks a durable identifier to a party that keeps it. Where it protects privacy, it usually does so by **abandoning accountability**: pseudonymous keys with no liable human behind them. Regulators want the second failure closed; users want the first one closed; and the arrival of autonomous AI agents — acting in fleets, at machine speed, across organisational boundaries — makes both failures worse simultaneously.

MintID is a **sovereign, permissionless proof-of-stake Layer-1** whose only job is to be the public trust layer for a different arrangement. **Accepted KYC issuers** — regulated organisations admitted by a compliance council and bonded in the native token — verify a person once and issue an **anonymous credential** into the person's own wallet. The holder then proves claims — *assurance grade ≥ A3*, *over 18*, *credential active*, *issuer accepted* — with **zero-knowledge presentations** that live for ten seconds, are bound to one verifier, one origin and one chain state, and never touch the chain. The chain stores only what must be public to make trust *checkable*: issuer and verifier registries, 30-second **status roots** committed by issuers, holder **self-revocation nullifiers**, staking, governance and evidence commitments. **No raw KYC, no personal data, no credential, no presentation and no per-holder record is ever written on-chain**, and live state grows with the number of issuers, verifiers and validators — never with the number of people.

0	10 s	30 s	108M
identities, credentials or presentations written on-chain	lifetime of a fully-bound zero-knowledge presentation	status-root heartbeat per issuer, fail-closed when stale	native tokens, a hard cap that is exact and immutable

The same machinery is inverted onto software: a verified human can mint **agent credentials** that prove *"I am operated by a KYC-verified, liable principal at grade ≥ G, within scope S, with ≥ F of self-custodied recourse behind me"* without revealing who that principal is and without two agents of the same person being linkable. Identity leaves the system only through two firewalled **disclosure rails**: a consensual, priced commercial rail that defaults to refusal, and a due-process dispute rail that settles most disputes with money before it touches identity.

01

The problem

> Identity today either leaks or cannot be held to account — and autonomous agents make both failures worse.

1.1 Two failure modes, one root cause

Every mainstream identity mechanism of the last twenty years — federated login, KYC onboarding at each service, age verification by document upload, payment-network identity signals — was built on the assumption that a human is at the keyboard and that *knowing who they are* is the price of *trusting what they claim*. The consequence is structural over-collection: the relying party learns a name, a date of birth, a document number, an email address, and keeps it, because that is the only shape the proof came in. Data-protection law (GDPR, and in the EU the coming eIDAS 2.0 wallet regime) now treats that over-collection as a liability, and the breach record of the last decade shows why.

The privacy-preserving alternatives that have shipped at scale — pseudonymous key pairs, self-issued decentralised identifiers, "verifiable credentials" whose issuer is whoever chose to sign — solved the leakage but gave up the thing regulators and counterparties actually need: a **liable, verified party** who can be reached when something goes wrong. A signature from an unaccountable key is not identity; it is a nickname.

Both failure modes share a root cause: **the proof of a claim and the identity of the claimant travel together.** MintID's design goal is to separate them cryptographically — the claim is provable, the identity is not — while keeping a *governed, bonded, on-chain-checkable* chain of accountability from every proof back to a regulated issuer.

1.2 Agents make it urgent

Autonomous AI agents break the "human at the keyboard" assumption outright. Industry estimates put the ratio of non-human to human identities in enterprises at roughly 50:1 (IBM/Omdia, 2024) and project more than a billion active agents by 2029 (IDC, 2025). Today's stack bolts agents onto service accounts, API keys and OAuth tokens, and fails on four fronts at once: no verifiable link to a liable party, unbounded blast radius when a credential leaks, no way to scope what an agent may do, and no recourse when it does harm. The commercial "know-your-agent" (KYA) rails that shipped across payment networks in 2025–2026 answer this by **revealing the principal's verified identity to the seller** — restoring accountability by re-introducing surveillance. MintID's thesis is that this trade-off is unnecessary: accountability can be *provable and funded* while identity stays private by default and is disclosed only by consent or due process.

1.3 What a solution has to be

The requirements follow directly from the failure analysis; they are the core requirement set of the MintID specification (Appendix A), summarised here:

- **A public trust layer that no one owns**: a sovereign, permissionless proof-of-stake chain with deterministic finality, so that issuer status, verifier status and governance are checkable by anyone and cannot be silently edited by any single party.
- **Nothing identifying on that layer**: no KYC documents, biometrics, names, credential serials, presentations or per-user events; state must scale with institutions, not people.

- **User custody**: keys and proof secrets stay with the holder; the holder can revoke their own credential without asking the issuer; recovery is by fresh KYC, never social recovery.
- **Accountable, bonded issuers**: only council-admitted KYC organisations issue accepted credentials; each posts slashable native-token collateral and can emergency-revoke its own registry alone.
- **Root-based, fresh status**: a compact commitment per issuer every 30 seconds, and fail-closed verification when it goes stale.
- **KYC-approved verifiers, bound to exact origins**: the relying party is itself accountable, and holders can pin whom they trust locally.
- **Perishable, fully-bound presentations**: ten seconds, single-use, bound to verifier, origin, policy and chain state, verified online against finalised chain state.
- **Hard-capped, transparent monetary policy**, and a **security-first release** with independent audits, adversarial testnets and a bug bounty before mainnet.

02

Design principles and hard invariants

> MintID is defined as much by what it refuses to do as by what it does.

Six invariants are treated as non-negotiable across the whole specification; every later section is a consequence of them.

Invariant 1 — The chain never learns who anyone is. No raw KYC, personal data, credential payload, credential serial, presentation, or per-holder issuance event is ever written on-chain.

Invariant 2 — State scales with institutions, not people. Live state grows with issuers, verifier origins, validators, status roots and governance records — never with holders, credentials or presentations.

Invariant 3 — Consensus is deterministic and self-contained. No network call, wall clock, randomness, hardware signer, remote service or foreign-function bridge runs during block execution; every validator reaches the same result from the same bytes.

Invariant 4 — Backing is provable, identity is not. A holder or agent proves *that* it is backed by an accepted issuer at a given assurance, never *which* person; siblings (two agents of one human) are unlinkable; the only link lives in the issuer's private escrow.

Invariant 5 — Disclosure is rail-segregated and, on the commercial rail, consensual. Identity leaves the system only through the consented commercial rail (default: refuse) or the due-process dispute rail; **no payment is ever a lawful basis for disclosing personal data**.

Invariant 6 — The value layer is transparent and hard-capped. 108,000,000 tokens, exact and immutable; the token never gains shielded or anonymity-enhanced transfer modes (the *AEC invariant*). Holder privacy lives in the credential layer, not the money layer.

Three further principles shape the engineering:

- **Reuse mature infrastructure; own only the identity-specific parts.** MintID does not write a consensus engine, P2P stack, staking system or light client; it is a Cosmos SDK application chain on CometBFT and spends its novelty budget on the credential protocol and the registries (§8.1).
- **A language boundary that is a security boundary.** Go owns consensus-visible state; Rust owns specialised cryptography; Python owns simulation, testing and automation and has no role in producing a block or holding a key (§8.2).
- **Bands before points; simulation before genesis; audit before mainnet.** Every economically load-bearing value is expressed as a percentage of the cap, ratified as a band with an argued prior, and fixed by an external simulation inside the band; every cryptographic component in a consensus path is a launch-blocking audit item (§10, §12).

03

Roles and the trust model

> Every party can be named, bonded or pinned — and none of them needs to learn who the holder is.

3.1 The actors

ROLE	MAY DO	MUST NOT DO
Holder (person, or a principal deploying agents)	Control credential secrets; prove predicates; self-revoke; request KYC re-verification	Delegate signing; publish credential identifiers; reuse a verifier challenge
Accepted issuer (regulated KYC organisation)	Perform KYC; issue and review credentials; publish 30-second status roots; emergency-revoke its own registry	Put KYC data on-chain; alter another issuer's credential; inspect holder presentations
KYC-approved verifier (relying party)	Request minimal claims; validate current chain state and the proof; enforce single use and expiry	Request unapproved claims; use wildcard origins; replay proofs; write proof activity to chain
Validator	Validate consensus and deterministic protocol rules; earn emission and fees	Access KYC documents; judge whether a real-world identity is genuine; store presentation data
Compliance council (threshold body)	Admit, suspend, revoke issuers and verifiers; adjudicate bonds; co-sign protected treasury actions	Individually move funds; approve a person's identity; change software or protocol parameters
Token governance (on-chain)	Protocol upgrades and versioned parameters	Operational compliance gates
Independent custodian / auditor	Co-sign reserve movements; attest custody and policy compliance	Act alone or replace council authority
Archive / index / premium-service operator	Serve historical data, status mirrors, paid verification services	Become a source of truth beyond chain proofs; influence consensus rewards

Validators — *miners*, commercially — never see a passport: KYC is performed by issuers, and validators and verifiers check cryptographic evidence only.

3.2 What sits where

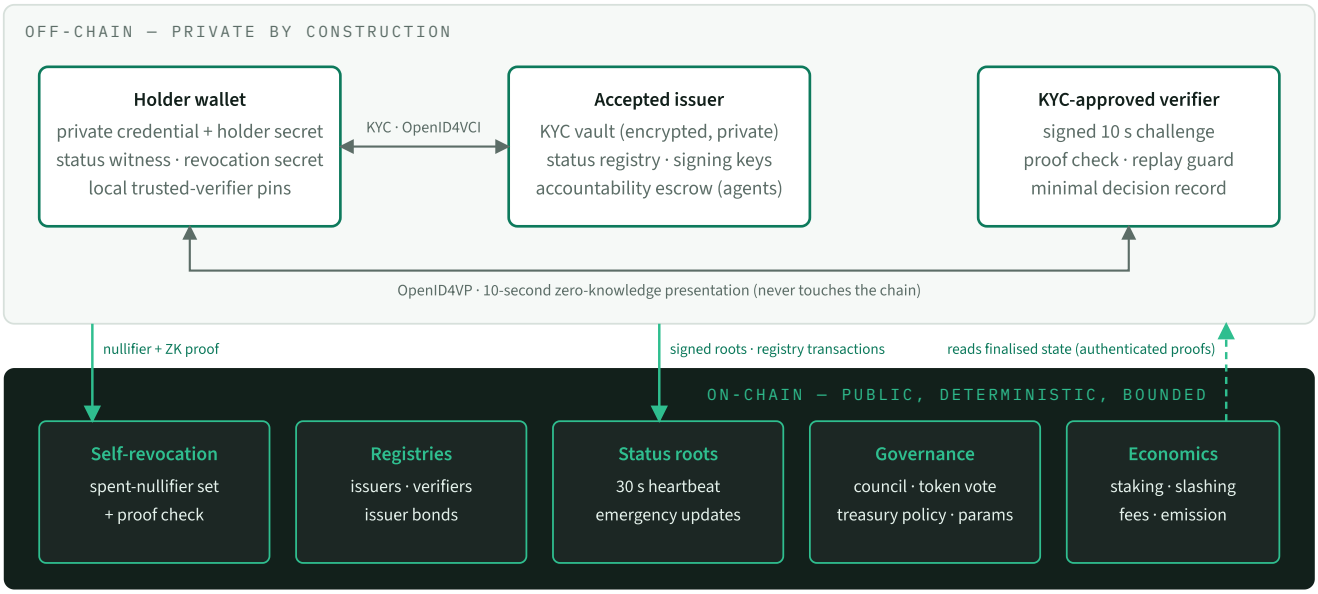

Figure 1 — Two planes. Everything that identifies a person stays in the off-chain plane; the chain holds registries, commitments and economics. Solid arrows write to chain (issuer roots, holder nullifiers); the dashed arrow is the verifier's read of finalised state.

Three data paths cross the boundary, and only three:

1. **Issuer → chain:** signed status roots (every 30 seconds; emergency roots immediately), registry transactions and bond operations. A root commits to the issuer's private status registry and says nothing about any individual credential.
2. **Holder → chain:** a self-revocation nullifier plus a zero-knowledge authorisation proof, submitted through a privacy relay. It reveals neither the holder nor a credential identifier.
3. **Verifier → chain (read only):** authenticated reads of finalised issuer and verifier state — never a trust-me RPC. Presentations never enter the chain; KYC evidence never leaves the issuer's vault.

3.3 Assurance grades

Every credential carries a private assurance grade `A1`–`A4`, set by the issuer within the tiers the council allows it.

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 audit requirements

A holder proves *grade* ≥ *requested grade*; the verifier learns the issuer and the policy result — never the exact grade, the expiry, the credential identifier or any attribute. Grades measure the *strength* of the verification, anchored on eIDAS and squared with ISO/IEC 29115 (§8.5); how *recently* the evidence was checked is a separate freshness band, never a date, so a verifier can ask for *grade* ≥ *A2* and *freshness* ≥ *F1* without either answer becoming a correlator.

04

The protocol at a glance: the life of a credential

> One credential, from its issuer's admission to its retirement, in six steps — each elaborated in §5–§7.

1. Admission. A KYC organisation applies to become an accepted issuer. The compliance council reviews it; the issuer registers its signing keys and a separately scoped *emergency status key*, is granted a set of permitted grades, and locks a native-token bond of at least the governed minimum. Only then does the chain let it become active; the bond is a condition of operation, not a formality.

2. Issuance. A person completes KYC with the issuer. The issuer stores the evidence, encrypted, in its private vault. Over OpenID4VCI it delivers an **anonymous credential** — signed blind, bound to a secret only the holder knows, carrying a private grade, a validity window and a private status reference — together with a status witness. The issuer records the credential's status in its private registry; at the next 30-second heartbeat it publishes a new signed status root, and once that root is finalised on-chain the credential is usable. **No transaction records that a credential was issued.**

3. Presentation. A relying party — itself KYC-approved and registered on-chain with exact HTTPS origins and request-signing keys — issues a signed, single-use challenge that names the verifier and its origin, states the claims it needs, records the finalised chain state it checked, and expires within ten seconds. The holder's wallet checks the verifier against its local trust pins, generates one zero-knowledge proof for all requested predicates, and returns it encrypted. The verifier re-checks finalised issuer and verifier state, applies a fixed set of acceptance conditions (§6.4), consumes the challenge so that it can never be replayed, and keeps a minimal decision record. **No presentation is broadcast, indexed or committed to the chain.**

4. Status and revocation. Every 30 seconds the issuer publishes a fresh signed root; the chain enforces the time bounds as consensus rules, and a verifier fails closed at 45 seconds of staleness. Issuer-side revocation is a change in the next root (≤ 30 s), or immediate through the emergency key. **Holder self-revocation** is a chain transaction that carries a one-time nullifier and a zero-knowledge authorisation proof, submitted through a privacy relay: it reveals neither the holder nor a credential identifier, and later presentations of that credential can no longer prove it unrevoked.

5. Renewal and recovery. Renewal is a new credential — from the same or a different accepted issuer — with fresh secrets, publicly unlinkable to the old one. Loss or compromise of the wallet has exactly one recovery path: fresh KYC re-verification and a brand-new, unlinkable credential. There is no social recovery and no custodial back door.

6. Agents. A verified human can additionally have the issuer *derive* agent credentials from the principal credential (§7): the human proves in zero knowledge that it holds an unrevoked credential at a sufficient grade, and the issuer mints a fresh, unlinkable agent credential carrying a backing assertion and a delegated scope. The chain sees nothing new — the agent credential is another entry under the same issuer's status root.

05

The chain

> A purpose-built application chain whose state is registries, commitments and economics — never people.

5.1 Why a sovereign application chain

The chain is deliberately *not* a smart-contract platform and *not* a rollup on someone else's settlement layer. Three requirements force sovereignty: the privacy invariants are **chain-wide** properties (state must be bounded and identity-free by construction, which a general-purpose VM cannot promise for its other tenants); the issuer council, verifier registry and status-root policy need **native capabilities with their own economics**; and the security budget of an identity trust anchor cannot be delegated to a market that prices tokens, not registries. The Cosmos SDK gives exactly this shape — custom native modules over a proven BFT engine — without writing consensus, networking or staking. Comparable systems informed the pattern: cheqd shows a Cosmos-ecosystem verifiable-data registry with AnonCreds support; Hyperledger AnonCreds supplies the credential model; Privado ID the issuer–holder–verifier triangle with local proof generation; the EU Digital Identity Wallet ecosystem the issuer / wallet / relying-party governance vocabulary and the OpenID4VC transports.

5.2 Consensus and validators

MintID runs delegated proof of stake on **CometBFT** through the **Cosmos SDK**: anyone can run a node, bond native tokens and register a validator; the top validators by capped bonded voting power form the active set, and the rest are standby validators or delegators. The genesis candidates are a capped active set of **60 validators**, growing toward 100 by governance once the fee-funded security budget supports it; block times of two to three seconds with a finality objective of about five seconds under normal conditions; a **5 % cap** on any validator's effective voting power; and a **21-day** unbonding period. The minimum self-bond is fixed after simulation.

The ten-second figure of the presentation protocol is a *holder-to-verifier* deadline; it does not mean that proofs are posted on-chain or that the chain finalises every verification.

Rewards. Each epoch's scheduled emission is shared among active validators in proportion to their capped voting power weighted by their signing performance; delegators share through the standard distribution path, validators charge their declared commission, and missed votes cost reward and, when sustained, lead to jailing. **No reward is ever paid for subjective "identity validation work"** — validators are paid to replicate a deterministic state machine, nothing else. Optional off-chain services (status mirrors, archives, paid verification services) earn direct fees that never touch consensus rewards.

Slashing is objective-evidence-only: double-signing → slash and permanent exclusion; sustained downtime → slash and temporary jail; alleged censorship → *no* slashing rule at launch (attribution is too subjective); issuer fraud → the separate issuer-bond process; verifier abuse → registration suspension. Validators are expected to run professional topologies — sentry nodes, separated keys, hardware-backed signing, monitoring and reproducible builds.

5.3 The economic-security launch gate

A young sovereign proof-of-stake chain is cheapest to attack at genesis. Because a consensus takeover here would compromise issuer status roots and registry integrity — not merely balances — MintID treats economic security as a **launch criterion of the same rank as the audit slate**. Mainnet genesis must not proceed until the network meets published, price-independent thresholds for (a) bonded stake, as a percentage of the maximum supply, held by *external* parties, (b) the number of independent validator operators, (c) a Nakamoto-coefficient floor, and (d) an

estimated attack cost in fiat computed by a published methodology. Ratified priors: **≥ 20 % of supply externally bonded, ≥ 25 independent operators, Nakamoto coefficient ≥ 7**, with the measurement method published before any offer opens; development and foundation accounts — and any delegation traceable to them — are excluded from all three. Bootstrap validators are recruited under fiat or stablecoin **service agreements** for the first 12–24 months (an instrument a professional operator can underwrite), continuing after the bootstrap window as the foundation-funded **operator floor program**, which pays for verifiable service — uptime, topology, monitoring — never for stake and never through a price oracle in a block-execution path. Whether the first 12–24 months warrant *rented* security (shared-security or Bitcoin-staking checkpointing services) is a mandatory simulation deliverable; adopting it is optional, evaluating it is not.

5.4 What the chain stores

MintID reuses the framework's standard modules — accounts, bank, staking, slashing, distribution, evidence, upgrades, consensus parameters and token governance — unchanged, and adds a small set of identity-specific capabilities:

CAPABILITY	PURPOSE
Issuer registry	Admission, authority state, signing and emergency keys, permitted grades, link to the bond, suspension
Issuer bonds	Lock, freeze, slash and release; remediation allocated per adjudicated case
Status roots	The latest signed root per issuer, consensus-enforced freshness bounds, emergency updates
Verifier registry	KYC-approved relying parties, exact-origin binding, request-signing keys, periodic revalidation, suspension
Self-revocation	The set of spent nullifiers and the verification of their proofs (enabled only after dedicated audits)
Compliance council	Threshold-vote proposals with evidence commitments and policy versions
Treasury policy	Reserve-movement commitments, timelocks, co-signature state, remediation claims
Protocol parameters	Versioned schemas, verification keys, freshness policy, fee and agent-layer parameters
Fees	Fee routing, burn accounting, the supply ledger and collection of the protocol cuts (§10.3)

Every record is deliberately small and institutional. An issuer is an opaque identifier, an organisation-level name, an authority state, its keys, its permitted grades, its current status root and its bond; a verifier is a commitment to its own KYC attestation (never the attestation itself), its keys and its registered origins — each one exact, with no wildcards. Issuer authority moves through a closed set of states — pending, active, under enhanced monitoring, suspended, and revoked as the terminal state — and any other transition is rejected by the chain. Activation requires a live bond at or above the governed minimum; a slash that exhausts the bond suspends the issuer in the same state transition, with no discretion and no grace window, and while an issuer is suspended the chain rejects its status roots, emergency ones included — holders and verifiers are then protected by the fail-closed freshness rule, which is strictly stronger than anything the issuer could publish. **No state table contains a holder's name, account, credential serial, KYC record, presentation, payload or proof history.**

Interfaces. Issuers manage their own keys and bonds; admissions and sanctions flow through the council (through token governance in the interim before the council is seated, with an explicit hand-over so that the interim cannot become permanent); status updates are relayed permissionlessly; and slashing is never a standalone transaction but

an action inside an evidence-backed council decision. Every integration reads chain state through authenticated store proofs verified against a light-client header — never by trusting a single RPC provider — and that path adds no new cryptography to consensus.

5.5 Status roots: how freshness becomes a consensus rule

Each issuer keeps a private status registry and publishes a small signed commitment to it at least every 30 seconds. Relayers batch many issuers' updates into a single transaction using standard, audited hash-tree constructions; they pay the fees and gain no authority — the only thing that authorises a root is the issuer's own signature, which the chain checks for every issuer in the batch. Signatures are domain-separated and pinned across implementations by shared test vectors.

Freshness is a **consensus rule, not a convention**. The chain rejects any root that is already expired, already stale, dated in the future beyond a small tolerance, or valid for longer than policy allows — without those upper bounds, a single compromised key could publish a root that reads as fresh forever. Verifiers treat a root older than 45 seconds as stale and **fail closed**; that is the only stale behaviour the protocol can represent. The heartbeat, the maximum age, the batch size and the emergency rate limits are governed parameters, the last two to be set from benchmark evidence.

A verifier rejects a proof if the issuer is not active, if the root is not the latest finalised one or has aged out, or if a newer emergency update or a suspension exists. **Emergency updates** let an issuer's separately scoped emergency key act alone on its own registry, with strict ordering, linkage to the previous root and per-block rate limits. The update is final after normal chain finality, contains nothing that identifies anyone, and patterns of emergency use are audited by the council, which can suspend or slash.

5.6 Why this stays small

At steady state, consensus state grows with active issuers, verifier origins, validators, revocation-tree roots and governance records — and *not* with people, credentials or presentations. Blocks record protocol transactions; ordinary nodes prune history and keep snapshots plus current state; archival nodes keep full history and evidence commitments. A billion credentials cost the chain the same as a thousand: one root per issuer.

06

Credentials, presentations and self-revocation

> One credential, many unlinkable proofs, and a way for the holder to revoke it without being identified.

6.1 The credential profile

MintID's primary credential is an **anonymous credential** in the Hyperledger AnonCreds lineage: blinded issuer signatures, holder binding through a secret that is never revealed, selective disclosure, predicate proofs, and non-revocation proofs that expose no correlatable identifier. The exact construction, library revision and proof system are frozen together with the other protocol values and independently audited before mainnet (§13, §14). Transport is **OpenID4VCI** for issuance and **OpenID4VP** for presentation, carrying a MintID credential-format profile. The OpenID4VC High Assurance Interoperability Profile (HAIP) standardises SD-JWT VC and ISO mdoc, *not* anonymous credentials; MintID uses HAIP as security guidance and does not claim HAIP interoperability unless a separate profile is implemented and certified (§8.3).

The credential itself is seen only by the holder and the issuer. Besides the issuer's signature it binds a holder secret that is never disclosed, a private assurance grade, a validity window, a private reference into the issuer's status registry and a commitment that later makes self-revocation possible — none of which a verifier ever learns.

6.2 Issuance and renewal

Issuance is the flow of §4, step 2; a credential is usable only after a root committing to its valid status has been finalised. Renewal is triggered by approaching expiry or a quality review: the holder performs new KYC or a review with *any* accepted issuer, receives a new credential with new secrets, the issuer publishes a later root, and the holder receives a new witness. A renewed credential must not publicly link to its predecessors — even when the issuer is the same.

6.3 The ten-second multi-claim presentation

A registered verifier opens a session with a signed, single-use challenge. The challenge names the verifier and the exact HTTPS origin it is registered for, carries a fresh random value, states the claims it needs as a policy — for example *assurance grade ≥ A3*, *over 18* and *credential active*, optionally restricted to particular issuers — records the finalised chain state the verifier checked, and expires within ten seconds. The holder's wallet answers with one zero-knowledge proof covering every requested predicate, optionally required to come from a single credential. The verifier learns only the policy results — *grade ≥ A3: yes*, *over 18: yes*, *active: yes* — and which accepted issuer stands behind them.

6.4 What a verifier checks

A verifier accepts only when **all** of the following hold. The check exists in exactly one implementation, and no client library may weaken it:

- the verifier and its exact origin are active on-chain, and the request was signed with a key registered for that origin;
- the challenge is fresh, has never been used, and was answered within its ten-second window;
- the presentation is bound to that verifier, origin, request key, challenge, policy and chain state;
- every requested predicate is proven;

- the issuer is active and its status root current, re-checked against finalised state immediately before the decision;
- the challenge is irreversibly consumed, so the same proof can never be accepted twice.

6.5 Verifier registration and holder-side trust

A verifier must first complete KYC with an accepted issuer, which produces a commitment to that verifier's KYC attestation; it then registers a request-signing key and its exact origins, proves control of those domains through standard web and DNS proofs (the chain stores only a digest), and the council approves. Origins are revalidated periodically, and wildcards are prohibited. On the holder side, the wallet keeps a **local** trusted-verifier store keyed to the verifier, its exact origin and its request key — there is no trust by name; any key rotation, origin change, KYC status change or enhanced-monitoring state requires fresh consent; and the holder's approvals are never on-chain and never visible to verifiers or validators.

6.6 Holder self-revocation

A naïve self-revocation would publish a stable credential identifier and defeat privacy. MintID instead lets the holder derive a one-time **nullifier** from a private revocation secret and prove, in zero knowledge, that it belongs to a valid credential currently committed by an accepted issuer — without revealing who the holder is or which credential it is. The nullifier and proof are submitted through a privacy relay; the chain verifies the proof and records the nullifier, and every later presentation of that credential must prove that its nullifier is absent — which it no longer can.

This is the only proof verified inside block execution, and therefore the most heavily gated component in the system: a published specification and circuit, deterministic and bounded verification in chain code, a test-vector suite, an independent cryptographic audit, cross-platform reproducibility and consensus-fuzzing gates, and adversarial testing for arbitrary revocation, replay, nullifier collision and credential linkage. It is **not enabled on mainnet until those gates pass**.

6.7 Recovery

There is no social recovery. When a wallet key is lost or compromised, the holder requests fresh KYC re-verification; the issuer may identify and revoke the prior credential in its private records; the holder receives a brand-new credential with new secrets, usable after the next finalised root. Old and new are publicly unlinkable.

07

The agent layer: verifiable human-backed agent identity

> The same machinery, applied to the software people deploy: agents prove who stands behind them without revealing who that is.

The agent and disclosure layers extend the same machinery to software. They are specified separately, subordinate to the core specification, and — this is the design's central engineering claim — **add no consensus module and no new proof type to block execution**. An agent credential is another credential under an issuer's status root; the only consensus-visible additions are governed parameters and fee routing.

7.1 Minting: derivation, not registration

A human **principal** who holds an unrevoked credential proves, in zero knowledge, that it holds it at a grade at least as high as the one requested for the agent — and, optionally, that its cap on active agents is not exceeded. The issuer then **mints** a fresh agent credential with its own secrets, a *backing assertion* and a *delegated scope*, and records the only link between agent and human in its private **accountability escrow** (distinct from the *financial* escrow below). The next status root makes the agent usable. There is no per-agent chain record, no public agent registry and no on-chain agent reputation score — by requirement, not omission.

The agent can then prove statements such as *human-backed*, *assurance grade* ≥ *G*, *scope includes this action*, *transaction limit at least X* and *credential active*. Scope is expressed in a structured vocabulary — what the agent may do, with whom, in which categories and places, when, and up to what value — and is signed at mint, so an agent cannot elevate itself. Two further attributes describe how the agent's keys are held and who operates it; a managed runtime is a signal, never a trust root.

The three invariants: backing is provable, identity is not; siblings are unlinkable — on chain, in presentations and across verifiers, with per-verifier pairwise pseudonyms; the only link lives in escrow. **Sybil resistance has a price**: a KYC gate, a per-principal cap on active agents, KYC freshness, and a mint fee with a recurring lease — all enforced privately by the issuer, without per-agent chain data.

7.2 Revocation and the kill-switch

Three levers: per-agent self-revocation through the same path as §6.6 (effective at the next finalised root); issuer status revocation (≤ 30 s, or immediate through the emergency key); and a principal-wide cascade under issuer policy, which must leak neither the principal nor the size of its fleet. A push channel may notify verifiers of revocations for convenience; it carries **zero security weight** — the root is the truth.

7.3 Funded accountability: self-custodied escrow with three exits

An agent can additionally prove *escrow coverage* ≥ *F* — that at least F of value — from the committed collateral set of **USDC, EURC and BTC** — self-custodied in the principal's own wallet under a smart-contract lock, stands behind it. The design refuses to become a deposit-taker: no fiat custodian, no third-party deposit, and no anonymity-enhanced coin in any collateral or reserve set. The lower funding bands are proof-only; **higher bands require the audited smart-contract lock** and are staged after the identity-only launch (§12).

Two engineering rules make the escrow safe to reason about. First, **the agent cannot move the money**: the secret an agent uses to present carries no authority over the lock, and discretionary withdrawal needs a separate release

factor that never exists in the agent runtime and that only the principal can exercise — directly, or through a consent policy that defaults to refusal. Second, **the lock has exactly three exits**: a value-conserving re-merge by the same agent, a payout to a claimant on a dispute verdict (under no party's key), or a return to the principal. There is no fourth exit; in particular, releasing escrow against an external payment settlement was considered and rejected.

A session may pledge a short-lived, bounded reservation of escrow capital, so the same funds cannot back two sessions at once, while each presentation stays on the ten-second hot path; collateral moves across chains only when it is locked, never inside a presentation.

7.4 Two disclosure rails, one firewall

The disclosure layer governs how identity flows *out*. Its invariant — **disclosure is rail-segregated and, on the commercial rail, consensual** — is enforced by construction: the two rails use different messages and different state machines.

The commercial rail exists because a merchant sometimes legitimately wants to *know* its customer. Here a relying party may *offer* a priced bid for one graduated item of information — a pairwise pseudonym, a single contact attribute, or full KYC. The bid is resolved by the principal's **consent**: a policy set at mint that can pre-approve small, bounded requests and otherwise escalates to the human, with **refusal as the default**. The bid is paid to the principal, minus a normative protocol cut and an optional issuer brokerage fee; refusal, timeout or error all mean that nothing is disclosed and nothing is paid. Delegation has a hard ceiling: a pairwise pseudonym is fully delegable; a single contact attribute only under a named allowlist with caps and expiry; and **full KYC is never delegable** — it requires live consent. A bid floor may be necessary but is never sufficient: *no payment is ever the lawful basis for disclosing personal data.* Consent choices are committed at mint in a perfectly hiding form; broadening consent is a paid re-mint, while narrowing or withdrawing it is **free** (GDPR Art. 7(3)). The product is deliberately **identification, not surveillance**: paying to identify one agent at one moment confers no right to recognise, re-contact or track the principal later — to know again, pay again — and no handle survives renewal.

The dispute rail cannot be vetoed by the subject — a defrauder would always refuse — and runs a **money-before-identity cascade**: **L0**, the escrow pays the claimant, with no identity and no council involved; **L1**, a minimal, banded fraud signal (a real KYC-verified human, a grade, a jurisdiction, a count of prior substantiated disputes), never personal data; **L2**, full unmasking under due process. Adjudication climbs a matching ladder: an optimistic claim-and-freeze backed by a claim bond; an anonymous determination by a **bonded arbiter selected by stake-weighted sortition**, paid by the losing party and self-executing against the lock (a binding third-party determination, not marketed as arbitration); and — only if both parties reject it *and* the arbiter certifies that a court is warranted — the judicial route, the only point at which identity is revealed. Claims carry a **zero-knowledge proof of harm** bound to the disputed session and capped at the proven coverage; objectively provable harms are decided by the proof, subjective ones by the arbiter. Every rung runs **off the consensus path**; the chain holds only arbiter-registry state and commitments to disclosure receipts — never their content.

On an L2 reveal the order is fixed: **the arbiter decides and certifies, the issuer releases, the council supervises — strictly after the fact**; no council approval is a precondition of a release. The unmasking threshold and quorum *(counsel)* gate the financial-recourse stage, not the identity-only launch.

Deliberately out of scope: self-derived minting and multi-hop sub-agent delegation (deferred); any per-agent on-chain record; a shared liability pool or insurance backstop; cross-counterparty clearing of value limits; a scope readable by every verifier; and any "bid buys identity" mechanic. The design also states its limits: correlation is intrinsic to *disclosure*, so unlinkability is not claimed for the commercial rail, and disclosure receipts are an audit trail, not a collusion detector.

08

Standards decisions and why they were made

> The choices architects and integrators ask about most, each with its alternatives and its reasons.

Each decision is stated as *decision — alternatives — rationale — status*.

8.1 A Cosmos SDK application chain, not a contract on an existing L1 or an L2

Alternatives considered: smart contracts on a general-purpose chain; a rollup; a permissioned ledger; a fork of an existing identity chain. *Rationale:* the privacy invariants (identity-free, bounded state) are chain-wide properties that a shared VM cannot promise; the registries, council, bonds and status-root policy are native capabilities with native token economics; an identity trust anchor should not rent its security budget from a market that prices somebody else's asset; and the Cosmos SDK is the framework whose whole point is custom native modules over a proven BFT engine — MintID writes nothing of consensus, P2P, staking or light clients. *Status:* ratified and implemented. There is no cross-chain dependency in the launch scope, and shared security is evaluated only as an option.

8.2 Engineering boundaries and primitives

Decision: Go owns everything consensus-visible; Rust owns specialised cryptography; Python owns simulation, test harnesses and automation, and has no role in producing a block, applying a transaction, computing a reward, verifying a consensus-visible proof or holding a key. In the consensus path the chain uses only standard, deterministic, widely audited primitives — Ed25519 signatures, SHA-256 hash trees in the Certificate Transparency style, and the Cosmos store-proof format for authenticated reads — plus the single, separately gated self-revocation verifier. *Rationale:* determinism. The chain must never call out to another process, a remote prover, a hardware signer, an RPC endpoint or an external database while executing a block, and **any foreign-function bridge into block execution is a launch-blocking determinism and cryptography audit item.** Off-chain services orchestrate the version-pinned cryptographic engine and never reimplement a cryptographic check; signing keys stay in hardware-backed signers. Authenticated reads let every integrator verify finalised state without trusting an RPC provider, and add no new cryptography to consensus. *Status:* frozen; the primitives are implemented and pinned by cross-language test vectors, and the batch construction is on the pre-mainnet audit slate.

8.3 Anonymous credentials, not SD-JWT VC or mdoc, as the primary profile

Alternatives: SD-JWT VC and ISO/IEC 18013-5 mdoc, the formats standardised by the OpenID4VC High Assurance Interoperability Profile and the EUDI Wallet architecture; W3C Verifiable Credentials with selective-disclosure signatures; DID-based issuer identifiers. *Rationale:* MintID needs *unlinkability across presentations and verifiers*, *predicate proofs* (*grade ≥ A3*, *over 18*) without revealing values, and *non-revocation proofs without a correlatable revocation identifier*. Salted-hash selective disclosure (SD-JWT) and mdoc reveal a stable issuer signature and are correlatable by design; they meet HAIP's threat model, not this one. Anonymous credentials in the AnonCreds lineage — blinded signatures, a holder secret, predicates, non-revocation — are the family that meets the requirements. Issuers and verifiers are identified by protocol identifiers and exact HTTPS origins, not DIDs: a DID layer would add resolution dependencies without adding a property the registries do not already give. *Consequences, stated honestly:* HAIP is used for security guidance only; **MintID does not claim interoperability with SD-JWT VC or mdoc credentials** unless and until a separate profile is implemented and certified — additional formats may follow once the primary profile is audited. *Status:* ratified; the exact construction, library revision and proof system remain open freezes (§14).

8.4 OpenID4VCI and OpenID4VP as transports — the EUDI-compatible envelope

Decision: issuance over **OpenID4VCI**, presentation over **OpenID4VP**, with a MintID credential-format profile inside them; the wallet is therefore built on the same protocols as the European Digital Identity Wallet. *Rationale:* these are the transports the eIDAS 2.0 relying-party ecosystem, its reference architecture and the wallet vendors are converging on; using them keeps MintID inside that ecosystem's governance vocabulary (issuer / wallet / relying party) and integration tooling while the credential *format* stays privacy-preserving. On top of them MintID adds what the transports do not carry: chain-native registration of KYC-approved verifiers, exact-origin binding and user-controlled verifier trust. *Status:* the request and challenge shape is implemented; full issuance and presentation semantics land with the proof core.

8.5 Assurance vocabulary: A1–A4 anchored on eIDAS, squared with ISO/IEC 29115

Decision: MintID keeps its own four-grade contract — grades are decided by the issuer from enumerated, authenticated checks and bounded by the tiers the council allows — and **anchors the grades on the eIDAS assurance levels, squared with ISO/IEC 29115**, a hybrid of the two approaches that were considered. The grade meanings of §3.3 are untouched; the external columns are mappings, not redefinitions:

GRADE	EIDAS	ISO/IEC 29115	NIST SP 800-63	AML POSTURE
A1	low	LoA2	IAL1	customer due diligence
A2	substantial	LoA3	IAL2	due diligence with biometric liveness (ETSI TS 119 461 baseline proofing)
A3	substantial + enhanced due diligence	LoA3^{+}	IAL2 + EDD	screening (sanctions, PEP, adverse media); evidence review at most yearly
A4	high	LoA4	IAL3	qualified identity proofing (ETSI extended proofing, notified eID at *high*, EUDI Wallet)

ISO's LoA1 (self-asserted identity) sits below MintID's floor and is out of scope; the third step is the anti-money-laundering enhanced-due-diligence step, not an additional eIDAS level. **Freshness is a second dimension, not a grade**: the issuer holds a freshness band (`F4` under 30 days, `F3` under 90, `F2` under 180, `F1` under 365), published as a band and never as a date so that it cannot become a linkable timestamp; a policy such as *grade ≥ A2 and freshness ≥ F1* is what "human-backed" means for ZadQ, the anchor product (§12.4). *Rationale:* eIDAS is what Europe's regulators, the EUDI Wallet and qualified trust-service proofing use first, and the AML Regulation points to it; ISO/IEC 29115 supplies the four-level backbone; and every framework reviewed tiers by *strength* while none attaches a validity period to a level — which is why validity moved out of the grade. The decision was informed by the first issuer implementation (§12.4), and the provider-neutral KYC interface did not change. *Status:* accepted in August 2026. The issuer-side KYC policy that applies the grades — requirements per type of operator (person or organisation), evidence reuse within the AML envelope, overdue re-verification handled as suspension rather than revocation, and consent withdrawal revoking everything it backed — is implemented behind the provider-neutral interface and awaits counsel sign-off *(counsel)*; until then the first issuer operates with a simulated KYC provider only.

8.6 Agent payments: compose with x402 and Web Bot Auth

Decision: agent payments integrate as a **transport profile**, not a protocol change. An x402 seller's payment-required response can state an identity requirement; the buying agent obtains a short-lived challenge from the facilitator, which acts as the registered verifier, and presents alongside its payment. The presentation binds to the facilitator's

origin; seller-specific requirements travel in the claim policy; and because presentations are perishable, a persistent "verified" badge cannot exist — an explorer instead publishes *its own* signed verdict, re-verified on a published cadence. The profile defines a three-step ladder — anonymous (payment only) → **Web Bot Auth** (identified software, signed HTTP requests) → Web Bot Auth plus MintID (identified software with an accountable human) — and can bind the bot's signing key into the agent credential. *Rationale:* x402, now under the Linux Foundation with payment networks and cloud providers as members, already places verification and settlement at the facilitator and offers an extension mechanism; composing with it and with agent-payment mandates ("bridge, don't fight") is cheaper and more credible than building a parallel rail. *Status:* draft profile with machine-readable schemas and tests, including a first **seller-side classical variant**: the seller's accountability is attested with conventional issuer-signed credentials and read back as a signed standing verdict — nothing in that exchange is a presentation or makes a zero-knowledge claim — to be superseded, not extended, by the zero-knowledge version. The dispute annex is reserved, and the profile freezes at the identity-only launch.

8.7 One verification pipeline, thin clients

Decision: the verifier's acceptance logic exists in exactly one implementation, offered identically embedded or as a service; client SDKs are **transport-only**; the ten-second lifetime is a constant in every binding, and no client exposes a "lite" or skip mode; challenges and request keys stay server-side. *Rationale:* per-language reimplementation of cryptographic acceptance logic is where identity systems historically leak — one core, one conformance suite. *Status:* accepted and implemented, with a versioned service contract and a first client SDK; further languages are demand-gated.

8.8 Regulatory posture: MiCA Title II as the only offer path, no anonymity-enhanced value layer

Decision: the token is designed as a MiCA Title II "other crypto-asset" (network security and utility; not an asset-referenced or e-money token; no claim on any entity's revenue). The public-sale tranche will be offered only through the full Title II process — crypto-asset white paper, notification to the AFM (the Dutch competent authority, following the Stichting's seat), offer, and admission to trading through a licensed crypto-asset service provider — and *nothing* is for sale outside it: no pre-sale, SAFT, points program or conversion promise; testnet tokens are worthless by construction. The value layer never gains shielded transfers (the AEC invariant), and no anonymity-enhanced coin sits in the reserve. *Rationale:* an identity network that wants regulated issuers cannot afford a token whose regulatory characterisation is contestable; MiCA Art. 76(3) and the EU AML Regulation make an anonymity-enhanced rail unadmittable; and putting holder privacy in the credential layer rather than the money layer keeps both regimes clean. *Status:* binding policy; execution is subject to counsel *(counsel)*.

09

Security architecture and threat model

> Issuers, verifiers, validators, relayers, the council and a compromised holder device are all assumed potentially malicious.

The chain must remain identity-free and deterministic under every one of them. The table maps each threat to its primary controls.

THREAT	PRIMARY CONTROLS
Raw KYC data breach *through the chain*	No raw KYC or credential payload on-chain by construction; issuer private vault; data-minimisation controls
Issuer fraud	Council approval; slashable native-token bond that gates activation; traceable issuer identity; audit commitments; suspension, enhanced monitoring and slashing with a public reason code and evidence commitment
Issuer key compromise	Only active keys verify; separately scoped emergency key; self-service rotation; council-adjudicated forced rotation; immediate root update; incident process
Validator collusion	Proof of stake; bounded, voting-power-capped set; BFT assumptions; slashing on objective evidence; the §5.3 launch gate; monitoring
Verifier impersonation / phishing	KYC-approved verifier registry; exact-origin binding, no wildcards; registered request keys; holder-side local trust pins
Presentation replay	Single-use random challenge; ten-second expiry; binding to verifier, origin, key, policy and chain state; durable replay guard
Credential correlation	Anonymous credentials, signature blinding, predicates; a fresh proof per session; no on-chain presentation log; per-verifier pairwise pseudonyms for agents
Wallet / key theft	Holder self-revocation; hardware-backed keys in the wallet; KYC re-verification as the only recovery
Stale revocation status	30-second roots; consensus-enforced freshness bounds; 45-second fail-closed staleness; online re-check of finalised state
Validator storage explosion	Root batching; no per-user state; pruning; snapshots; archive separation
Proof-system bug	Formal specification; deterministic verifier; test vectors; multiple independent audits; bug bounty; self-revocation disabled until audits pass
Council capture	Threshold quorum (recommended 5-of-9); independently selected members; one-for-one seat rotation; public commitments; custodian co-control; the council cannot change software
External reserve misuse	Council threshold **plus** an independent custodian signature; on-chain commitment; reason code; timelock; restricted-use policy; external audit
Sybil agent armies	Human KYC gate; per-principal cap; mint fee and lease; KYC freshness — enforced privately, with no per-agent chain data
Principal / sibling de-anonymisation	Sibling unlinkability; pairwise pseudonyms; no on-chain presentation log; escrow disclosure only by due process, with evidence commitments

THREAT	PRIMARY CONTROLS
Scope inflation	Scope signed at mint; no self-elevation
Accountability-escrow abuse	Rail segregation by construction; default-refuse consent; arbiter-decided, issuer-released, council-supervised-after-the-fact reveal; perfectly hiding consent commitments; launch-blocking governance gate
Coercion into disclosure	No bid is ever a lawful basis; full KYC never delegable; refusal is free and leaves no trace

Eight engineering rules are non-negotiable across all of it: consensus code never makes network calls, reads wall-clock time outside block time, or uses non-deterministic APIs; a validator validates every transaction from its bytes and current state alone; presentation and KYC flows stay off-chain; identifiers, schemas, verification keys and policy semantics are versioned; root and schema changes require a migration policy and testnet rehearsal; no API is trusted because it returns well-formed data; no external reserve is treated as instantly available or as token backing; and mainnet launches with conservative resource limits that grow only on measured evidence.

Release discipline. No mainnet before: two independent chain and protocol security audits, a specialist cryptography audit, an infrastructure penetration test, an independent token and economic review, a public adversarial testnet with independent validators, issuers and verifiers, a funded bug bounty, and an incident-response exercise. Any new cryptography in a consensus path is a launch-blocking audit item. If the economic review objects to any value frozen before genesis, genesis does not proceed on that value: it is reopened, revised with the finding on record, and frozen again.

10

Token economics

> A security asset with a hard cap, a declining emission and a burn — and no price in this paper.

> **Status of every figure in this section.** The maximum supply is frozen. Everything else is either a *ratified band* (binding: points may only move inside it, by governance) or a *ratified prior* (an argued working point that the independent pre-genesis economic simulation validates or argues against, inside the band, before genesis). No price is stated anywhere in this paper; the offer price, if an offer is made, will exist only in the MiCA Title II white paper. This section is disclosure, not an offer.

10.1 What the token is for

The native token is the network's **economic-security asset**: validator bonding and delegation, issuer bonds, transaction fees, registry leases, and governance-weighted operations. It is deliberately *not* a stablecoin, not backed by any reserve, not price-supported, not dividend-bearing, and not a claim on any entity's revenue. Value accrues to holders through exactly one passive channel — the burned share of protocol flows — never through distributions. It is a **launch vehicle for network security, not a scarcity instrument**: scarcity is carried by the burn, not by the emission curve.

10.2 Supply and emission

QUANTITY	DEFINITION
Maximum supply	**108,000,000 tokens** — an immutable hard cap, exact (frozen)
Minted at genesis	The development and public-sale allocations, in a single genesis event
Reward pool	The remainder — prior **64,800,000 (60 %)** — usable only for validator emission
Emission per epoch	The scheduled, declining emission, never more than what remains in the pool

The cap is counted against the total ever minted, not the circulating supply: **burns never re-open mint headroom** — a chain that has burned heavily is *further* below the cap, never entitled to mint again. Emission is paid out per block and rounds down, so realised emission can only undershoot the schedule. The reward pool has exactly one use — validator emission — and no other draw on it exists or may be created.

The emission curve must exhaust the pool exactly. The published curve's cumulative emission must *equal* the reward pool: the cap is real, no tranche may be unreachable by construction, and every supply or fully diluted figure in any offer document equals the cap without correction factors. Ratified prior: **initial emission of 3 % of the maximum supply per year (3.24 M tokens), decaying by ×0.95 per year** — a geometric series that sums to exactly 64.8 M; roughly 40 % of the pool is emitted in the first ten years and about 79 % within thirty, and the curve is strictly declining throughout.

The network can be **net deflationary** over a period only when burns exceed emission in that period; no perpetual deflation is promised. Burn sources: issuer admission, renewal and registry-lease fees; a defined portion of status-publication fees; a defined portion of issuer-bond and validator slashes; the burned share of the agent mint and lease and disclosure protocol cuts; and an optional, governance-configured protocol-state rent. There is no recurring tax on validator stake to force deflation — that would undermine the security it is meant to protect.

10.3 The five economic flows and the consensus firewall

FLOW	WHAT IT IS	WHERE THE MONEY GOES	TOUCHES THE TOKEN?
A — Agent mint & lease	A per-agent mint fee and recurring lease on a governed schedule	Issuer revenue; a **normative protocol cut** with a burned share	**Yes** — the one protocol revenue line that scales with the agent population
B — Issuer registry	Admission, renewal and active-registry lease; the slashable issuer bond	Fees partly burned; slashes split between burn and remediation	Yes — scales with the (deliberately small) issuer set
C — Consensual disclosure	The issuer-brokered disclosure-bid rail	Bid to the principal; optional issuer brokerage; a **normative protocol cut** (a percentage plus a fixed per-disclosure floor) with a burned share, collected on aggregate settlement	**Yes** — no per-holder or per-presentation state is created
D — Dispute / arbiter market	Claim bonds, slashable dispute bonds, arbiter fees paid by the losing party	Off-chain service revenue to arbiters; a forfeited bond compensates the counterparty, and only the protocol cut is partly burned	**No** — must never influence consensus rewards
E — Premium verification services	Paid verification services, mirrors, archives	Direct fees to the operator	**No** — and **basic verification is free by design**

Flows A and C carry the *only* protocol cuts. The firewall on D and E is not a gap to be closed later: it is what keeps verification free (the adoption lever), presentations off-chain and consensus neutral. **No usage tax exists or is planned.**

Ratified bands and priors for the cuts:

PARAMETER	RATIFIED BAND	PRIOR
Flow-A protocol cut	8–16 %	12 %
Three-way split of each A/C cut — burned share	20–40 %	30 %
— validator-directed share	35–55 %	45 %
— foundation share	15–30 %	25 %
Ordering rule	validator-directed ≥ burned share until fee-funded security coverage is demonstrated on realised adoption (“security first”)	—
Agent mint and lease prices	set before genesis from the simulation; the lease prior is a competitive *ceiling*, and a cross-issuer anti-sybil floor is enforced by admission policy	point from simulation
Flow-C fixed per-disclosure floor	band ratified before genesis	point from simulation

The **foundation share** is earmarked — first the operator floor program (§5.3), then foundation operations, with a public annual report — and is **never distributed pro rata to token holders**. The alternatives were rejected on the record: an on-chain subsidy indexed to operating costs would need a fiat oracle near consensus (a violation of the determinism boundary); direct distribution to holders was rejected outright (dividend-like distributions strengthen a

security or asset-referenced-token characterisation); and raising issuer fees would sharpen the broker conflict that the disclosure safeguards exist to contain.

10.4 The post-emission security budget

The emission schedule is finite. After the pool is exhausted, the validator budget is the fee market: transaction fees plus the validator-directed share of the flow-A and flow-C cuts. The pre-genesis simulation must model this under published adoption scenarios and state whether it sustains the target set. If it does not, a **succession ladder is committed, in order**, and no step may be skipped: (i) governed adjustment of cut points inside the ratified bands; (ii) activation of the protocol-state rent, redirected pro rata to validators; (iii) supplementary shared security. **A post-cap mint is prohibited by construction and permanently excluded from the ladder.**

10.5 Bonds, remediation and the reserve

Every active issuer locks a bond of at least the governed minimum, sized by the simulation against a **published fiat-denominated deterrence target** — the bond must exceed the modelled gain from the cheapest slashable violation — and revisited annually. Each slash splits into a burned share and a **remediation share** held per adjudicated case for harmed parties; unclaimed remainders move, after a governed period, to an earmarked compartment whose only two exits are a late claim bound to the original case or a burn — the body that adjudicates slashes must never profit from unclaimed remediation. Separately, a small off-chain **BTC remediation and critical-security reserve** — controlled by a council threshold *plus* an independent custodian's signature, with on-chain commitment, reason code, timelock and post-disbursement report — exists only for proven issuer-fraud remediation and critical incidents. It is not token backing, not a bridge, not collateral, and never used for operations, rewards, grants or price support.

10.6 Genesis allocation

15 % development and team, vested and locked

25 % public-sale tranche, only via MiCA Title II

60 % reward pool, validator emission only

1 genesis event — no discretionary mint after it

COMPARTMENT	RATIFIED BAND	RATIFIED PRIOR	PURPOSE
Development / team	15 %	**15** % (16.2 M)	Developer compensation (four-year continuous vesting, one-year cliff, native vesting accounts), operations treasury, ecosystem and tester grants — all in labelled sub-accounts
Public-sale tranche	20–25 %	**25** % (27 M)	The MiCA Title II public offer that finances the *network launch* — audits, validator bootstrap, counsel, Foundation, custodian, runway
Reward pool	≥ 60 %	**60** % (64.8 M)	Validator emission — the security budget remains the dominant allocation

Rules that bind wherever the points land: all points are percentages of the cap, fixed by the simulation inside the bands, and a *lower* recommendation governs (a sale point below 25 % returns to the reward pool, and the raise floor is re-verified); there is a **single genesis event** and no post-genesis discretionary mint; a Genesis Allocation Statement lists every genesis address, label, percentage, vesting, custody and whether the no-delegation / no-vote rule applies, and its hash is committed in the genesis file so that anyone can reconcile block 1 against it; development and foundation accounts **must not delegate and must not vote** while vesting or lock-up applies, and are excluded from the launch-gate arithmetic; a 12-month post-genesis lock-up (extendable) binds development tokens to their labelled addresses; treasury and grants accounts sit behind a multisig with an **independent co-signer** and a timelock

— one custody standard for the token treasury and the BTC reserve alike; the independent economic review covers the allocation, the sale plan and the use of funds *before* they exist; and every related-party role of the implementing group is disclosed as such (§11.4).

Offer discipline (binding policy; execution subject to counsel). Exactly one document will ever be a MiCA crypto-asset white paper, and this is not it. The policy commits to a single fixed price — no tiers, no discounts, no repricing — and to a **raise floor of $4.0 M net of statutory withdrawals** that is invariant: if an offer closes below it, the project **re-stages instead of under-funding** (deferring spend, resizing the validator bootstrap, narrowing scope — never an under-audited launch), and the condition-triggered genesis of §12.5 exists precisely so that the floor never has to be lowered. Every other offer term — subscription currencies, eligibility, the treatment of unsold tokens — will be set only in the MiCA white paper *(counsel)*. Stake bought in the sale counts toward the launch gate. Externally, the token is never described as a financing vehicle for the implementing group: the offer finances the network's launch under a published budget, and the group's pre-genesis spend is its own R&D, disclosed as a related-party fact and never reimbursed (ADR-0016).

11

Governance and the MintID Foundation

> Software is governed by token holders, compliance by a threshold council, and the whole by a Dutch foundation.

11.1 Two organs, never conflated

Token-weighted governance (on-chain) owns the software: protocol upgrades and forks, consensus parameters, and every versioned protocol parameter. **The compliance council** is an operational body: it admits, suspends, reinstates and revokes issuers and verifiers, adjudicates bonds, co-signs protected treasury actions, escalates and clears enhanced monitoring, freezes and unfreezes bonds, and forces key rotation for a lost operator. It never changes software or parameters; its own size and threshold are governance parameters. On the dispute-path identity reveal its role is supervisory and *after the fact* only (§7.4).

The council votes on-chain with its own deterministic threshold (recommended initial configuration **5-of-9**); every action commits a reason code and an evidence commitment (sensitive evidence stays off-chain), a policy version and an effective time; a council key rotation is a one-for-one seat swap, and approvals from the rotated-out seat lapse on open proposals so that a seat is never represented twice. External-reserve movements additionally require the independent custodian's signature. Token governance is live from genesis; the council takes over the operational gates when it is seated, in the same proposal that seats its first membership.

11.2 The Foundation

The steward of the network is a dedicated **MintID Foundation**, a **Netherlands** *Stichting* seated in Amsterdam and associated with the project's public domain (its Chamber of Commerce registration details are published when it is activated). **Until the public-offer route is chosen, the Foundation is dormant** — no treasury, no contracts, no seats, no services received — and MintID is developed and operated by its first implementer, the CONECTIA group, on a private substrate under an Apache-2.0 licence (ADR-0014, ADR-0016); the financing route is decided at the Gate-1 go/no-go. Once activated, it performs the genesis, holds the treasury and the lock-up commitments, publishes the Genesis Allocation Statement, anchors governance neutrality and — as offeror — fixes the home Member State under MiCA Art. 3(1)(33): the notification authority is the Dutch **AFM**. Governance is staged, and the staging is itself a ratified decision: incorporation with the founder as sole director is accepted **for the dormant, pre-genesis phase only**; before any white paper is signed, the board seats independent members, statutory locks take effect (a cap on unrelated activity, the founder below a board majority, and a devolution clause), and the **compliance council is constituted as an organ of the Stichting**. The statutes carry the amendment path and the hooks for these organs. The reason recorded is not optics: white-paper liability under MiCA Art. 15 is personal and cannot be limited, so expanding the board before signature is a risk requirement. The Foundation has no *protocol* revenue role — it charges nothing for issuance or verification; its income is the foundation share of protocol cuts (which funds the operator floor program and its own operations under a public annual report), grants and, if a sponsorship-seat programme is opened, published dues that buy voice and visibility and never a vote (ADR-0017). A Stichting has no shares or members, and no debt relationship exists between it and the implementing group in either direction.

11.3 Transparency and post-production review

A public transparency report carries issuer and verifier states, council actions, bond slashes, root-liveness statistics, reserve-policy activity, movements from labelled genesis addresses, vesting releases, grants, validator service-agreement counterparties and related-party flows — **never holder data**. Required reviews: a quarterly issuer-quality

and false-issuance review; a quarterly verifier privacy and domain-control review; continuous validator concentration, liveness, equivocation and partition monitoring; an annual external cryptography and application-security audit, plus audits on any proof-system or consensus upgrade; and an annual review of the grade definitions, validity periods, bond values and remediation policy.

11.4 Related parties, disclosed

MintID was conceived and is being built by the **CONECTIA group** (CONECTIA OÜ, Estonia, and a wholly-owned Spanish operating company). That group is a **related party** in every investor-facing document, and this paper follows the same rule rather than claiming an independence it does not have:

ROLE	HELD BY	FLOWS TO DISCLOSE
Founder	Marc Molas	Founder of MintID; beneficial owner of the CONECTIA group; founder-director of the Stichting at incorporation. Recused from every Foundation decision that concerns the group; the founder-separation commitments of ADR-0017 apply on activation
Development company	Spanish SL (100 % CONECTIA OÜ)	Development allocation; operating budget from proceeds. Pre-genesis work is CONECTIA's own R&D at its own cost: no advance, no reimbursement, no debt in either direction
First accepted issuer and first registered verifier	CONECTIA group	Mint and lease fees, disclosure brokerage — on the same bond and admission terms as any issuer
Reference SDK / premium-services operator	CONECTIA group	Direct service revenue, firewalled from consensus rewards
Anchor-product operator (ZadQ)	CONECTIA group	x402 seller-verification service revenue (fiat); base verification stays free
Operator of ZadQ's private MintID deployment (§12.4)	CONECTIA group	No on-chain money flow beyond nominal fees and the issuer bond; operated for the anchor company under a disclosed related-party arrangement; reported as a pilot and excluded from the traction criterion
Pre-genesis R&D funder	CONECTIA group	An internal pre-genesis budget of €400,000 spent by CONECTIA on MintID as its own R&D; disclosed in amount; no services on credit to the Foundation, no reimbursement, no claim on proceeds
Steward	MintID Foundation (Stichting)	Dormant until activation; then treasury custody, grants, sponsorship dues and the foundation share; no protocol revenue role; no debt with the group

The first-implementer posture is **privilege-free and verifiable on-chain**: the group's issuer passes the same council admission, posts the same slashable bond and publishes the same 30-second roots as any other issuer, from block 1. Related-party service agreements follow the genesis policy's related-party approval circuit; the independent economic simulation contractor and the pre-mainnet reviewers may hold no position in the project and have no commercial relationship with the group. The independence that *is* claimed is methodological — every figure sourced, every model re-runnable by a third party — and the independent economic review remains the only genuinely arm's-length opinion.

12

Launch plan: staged, gated, condition-triggered

> Each stage opens only when its published gate is met; if a minimum is not reached, the plan re-stages.

12.1 The master rule

MintID launches in stages, each behind published gates, under one rule that overrides every schedule: **if a minimum is not reached, re-stage — never launch under-funded, under-audited or under-secured.** The identity-only network ships first; the financial-recourse rail activates on organic demand; genesis is triggered by conditions with a target date, not by a calendar. The reasons are on the record: a dispute rail paid only by dispute losers has no supply side at zero volume (staging closes the arbiter cold start without subsidies); coupling the launch to a multi-circuit zero-knowledge audit would let one slip block everything; and the competitive window for privacy-preserving agent identity is measured in quarters, not years.

12.2 The stages

STAGE 0
Pre-genesis operation
Frozen specifications, internal testnet, Foundation, first issuer and verifier, simulation and counsel
now → 2027

STAGE 1
Public offer
MiCA Title II white paper, AFM notification, offer, admission to trading
2027

STAGE 2
Identity-only mainnet
Credentials, presentations, status roots, free verification, 60 validators
H2 2027 → 2028

STAGE 3
Financial recourse
Escrow, disputes, arbiter market, consented disclosure
2028 → 2029

STAGE 4
Maturity
Many issuers, a growing agent population, active set toward 100
2030 → 2031

STAGE	TARGET WINDOW	SCOPE	EXIT / GATE
0 — Pre-genesis operating mode	now → Gate 1	The **first implementer runs the first issuer, the first verifier and the ZadQ pilot in production on a private MintID substrate (§12.4; ADR-0014, ADR-0016)** while the distributed network is built; internal testnet, frozen specifications, proof core in development (verification fails closed until it ships), code Apache-2.0; the Foundation seated in Amsterdam and dormant; funded by the first implementer as its own R&D, never reimbursed	Gate-1 go/no-go on the financing route · on the public route: simulation delivered, counsel workstreams, trademark filed, Foundation board and statutory locks in place
1 — Public offer	2027	MiCA Title II white paper → AFM notification → public offer of the sale tranche (25 % prior) → admission to trading	White paper cleared by counsel and notified; board and locks effective before signature; bank and custodian onboarded; disclosure pack published; **$4.0 M net raise floor** met, else re-stage

STAGE	TARGET WINDOW	SCOPE	EXIT / GATE
2 — Identity-only mainnet	genesis target H2 2027 → 2028	Issuance, zero-knowledge presentations (human-backed, grade, scope), status roots, free verification, verifier registry, council live, a genesis validator set of **60**; **no escrow or disputes yet**; ZadQ on mainnet	Audit slate clean (including the economic-review objection loop) · §5.3 thresholds met in measurable units · genesis values frozen · Genesis Allocation Statement published
3 — Financial recourse and disputes	2028–2029	Self-custodied escrow locks (USDC, EURC, BTC), session reservations, the L0/L1/L2 cascade, the arbiter market, consented commercial disclosure	Published organic-demand activation criterion · zero-knowledge circuit audit package · escrow-contract audits · stablecoin depeg parameters · unmasking governance *(counsel)*
4 — Maturity	2030–2031	Multiple issuers, a growing agent population, the active set grown toward 100 by governance, the post-emission transition planned on the committed ladder	Continuous: transparency reporting, annual audits, gate metrics monitored

Windows are targets, not commitments: each stage opens only when its gate is met. Underneath the stages sit seven build phases: preconditions and freeze; the chain foundation (done); issuer, council, bond and status capabilities (done on-chain, with the status plane and a first issuer service off-chain — issuance waits on the proof core); the verifier registry and verification core (done, except the proof engine); anonymous credentials and issuance (open — the proof core); holder self-revocation (open, audit-gated); and the public adversarial testnet and launch gate.

12.3 The eIDAS clock

The launch target is anchored to an external date: **24 December 2027**, when the eIDAS 2.0 Art. 5f relying-party acceptance obligations for the European Digital Identity Wallet take effect, and EU sectors will be choosing verification stacks. Because MintID's wallet speaks the same issuance and presentation protocols by construction, being live and audited before that date is the strategic reason the critical path is sequenced as it is: the external simulation and counsel outputs are sequenced to make a 2027 offer window and a genesis in the second half of 2027 achievable.

12.4 The first implementer and the anchor product

An identity network has a two-sided cold start: no issuer will bond capital for a chain with no verifiers, and no verifier will integrate against a chain with no credentials. MintID answers it with a **first implementer** and an **anchor product**, **ZadQ**, both disclosed as related-party positions (§11.4).

The first implementer — the CONECTIA group — operates the network's first accepted issuer, behind the provider-neutral KYC interface, and its first registered verifier, on **undifferentiated terms**: same council admission, same slashable bond, same 30-second roots, verifiable on-chain from block 1. It also builds and operates the reference wallet and verifier implementations under a permissive licence. Because the dogfooding is verifiable on-chain, the position becomes an argument rather than a risk.

ZadQ, the anchor product, is an **x402 seller-verification service** and the first product built on MintID: a seller that charges per request in stablecoins proves, through the standard verification path, that an accountable, KYC-verified person or organisation stands behind its payment address and resource — without revealing who. Explorers and buying agents consume the signal for free, and a registered verifier re-checks sellers on a published cadence and

publishes its own signed verdict. It requires **no protocol change**: it is an ordinary issuer-and-verifier pairing. It targets sellers and agents with meaningful economic volume and regulated business counterparties, and it sells **privacy, funded recourse and compliance by design — not cost savings**. Its first version runs **before the audited proof core, with conventional issuer-signed attestations on the seller side** — the seller's endpoint is public, so unlinkability adds nothing there — and **makes no zero-knowledge claim for that rail**; buyer-side agent credentials remain gated on the audited proof core, with no interim non-zero-knowledge mode. ZadQ runs as the group's own product line (zadq.app), with success criteria defined in advance and measured.

ZadQ starts on a private deployment, not on the network. Because no distributed MintID network exists yet, ZadQ's first attestations run on a **private, self-hosted installation of the same MintID software, operated by the first implementer**. MintID is *operated, never forked*. That deployment is **not the MintID network and is never described as a launch**: the public testnet remains the distributed target, and **nothing measured on it counts toward the traction criterion** — it is reported as a related-party pilot (§11.4, §12.5). The path changes one thing at a time: first the proof form (the standard zero-knowledge presentation, once the audited proof core and a holder client exist), then the topology (the distributed network). Custodial or facilitated payment addresses — where a third party holds the seller's key — are outside the first version *(counsel)*, because the first attestation has exactly one meaning: *the key holder is the accountable party*.

Neither role carries a protocol privilege, and neither is a substitute for the network: the traction criterion below counts *independent* verifiers only and treats the group's own deployment as context.

12.5 Condition-triggered genesis and the traction criterion

Genesis has a **target** (H2 2027) and a **trigger**; the trigger, all three parts of it, decides:

1. **The published composite traction criterion** — at least *N* attested x402 sellers with at least *R* % retained after 90 days; at least *M* active, independent registered verifiers, the promoter group excluded; and both sustained for two consecutive quarters. *N*, *M* and *R* are calibrated by the external simulation and published in measurable public units before any offer opens. Every component is registry-based or aggregate, because the protocol forbids central counting of verifications, and activity on ZadQ's private deployment (§12.4) is excluded outright and reported as a related-party pilot.
2. **An acceptable raise window** — the offer clears the invariant $4.0 M net floor.
3. **The simulation and counsel outputs delivered** — the work that fixes the genesis values.

If any part is missing, the **pre-genesis operating mode continues** — private substrate, first-implementer issuer and verifier, ZadQ, dormant Foundation — funded by the first implementer as its own R&D, with no operating loss and nothing to reimburse; a failed or below-floor raise re-stages rather than lowering the floor. In that mode, product and service marketing is unrestricted, while token marketing is prohibited until a white paper is published (MiCA Art. 7).

12.6 The pre-genesis economic simulation

Every economically load-bearing value is a band with a prior, so the external simulation is the hinge of the plan: without it there is no genesis, no allocation and no white paper. It is external by requirement, budgeted in the low-to-mid six figures and deliverable within one quarter of commissioning. It fixes the emission, allocation and cut points, the launch-gate thresholds, bond sizing, the post-emission security budget and the traction and recourse-activation criteria, with a protocol P&L under adverse, base and growth scenarios — the adverse case shown, not averaged away. Its market figures are primary-sourced, its model is re-runnable by the independent economic review, and changes to fixed inputs go through a recorded objection procedure.

13

Implementation status (September 2026)

> What exists today, stated at the level of capabilities, so that no reader mistakes specification for delivery.

MintID is a greenfield protocol under active construction. The table below is a candid, capability-level statement of what exists at the time of writing. Since the first draft of this paper (August 2026) the private anchor deployment of §12.4 has come together; the proof core has not.

COMPONENT	STATUS	WHAT IT MEANS
Chain (Go, Cosmos SDK)	**Implemented** — self-revocation pending	The application chain with the issuer, bond, verifier, status, council, treasury, parameter and fee capabilities; deterministic primitives pinned by cross-language test vectors; determinism and state-proof tests. The self-revocation capability is not implemented yet and is gated on its own audits.
Proof core (Rust)	**In development**	No anonymous-credential cryptography ships yet. Until the audited proof core lands, every presentation is refused by design (fail-closed). The construction, proof system and library revision are not yet frozen.
Verifier service	**Implemented** — proof engine pending	The full acceptance pipeline, single-use challenges, authenticated chain reads and an audit trail behind a versioned service contract, plus the standing read used by ZadQ's classical seller rail.
Issuer services	**Status plane and first issuer service implemented** — issuance pending the proof core	Encrypted status registry, 30-second root publication and relaying, a hardware-signing interface, the provider-neutral KYC interface with a simulated provider, and conventional seller attestations for the ZadQ pilot.
SDKs	**First client available**	A transport-only client and an agent-tooling integration over the verifier service; further languages are demand-gated.
Test harness	**Implemented**	Issuer, holder and verifier simulators; end-to-end, adversarial and transport-profile tests; an end-to-end test of the ZadQ integration.
Deployments	**Internal only**	An internal testnet (not public) and ZadQ's private deployment (§12.4). **There is no mainnet, no genesis, no token price and no public validator set.**
Specifications	**Frozen**	Core, agent-identity and disclosure specifications, genesis-allocation policy, token-sale plan and simulation scope — frozen in August 2026 and available on request, with decision records and improvement proposals maintained alongside.
Manuals	**Published**	Issuer, Verifier & SDK, Holder and Validator manuals.

14

Open parameters, open decisions and risks

> What still has to be fixed, what is deliberately left open, and what could go wrong.

14.1 Parameters that must be finalised before genesis

1. Genesis distribution points, the emission curve, fee splits, burn percentages, the protocol-cut and three-way-split points, and the per-disclosure floor — inside the ratified bands, from the simulation.
2. The §5.3 economic-security thresholds in measurable units, and the outcome of the rented-security evaluation.
3. The minimum issuer bond, slashing ranges, remediation allocation and lock periods.
4. Council membership selection, quorum, tenure, key-custody policy and appeal process.
5. Verifier registration cost, an optional verifier bond, revalidation cadence and abuse-remediation policy.
6. The active validator cap, minimum self-bond, voting-power cap, unbonding period and jailing thresholds.
7. **Proof-system selection and the exact anonymous-credential construction and revision.**
8. The status-root maximum age and emergency rate limits, after benchmark evidence.
9. Supported jurisdictions, KYC policy standards and data-retention policy. The issuer-side KYC policy exists (§8.5) and awaits counsel sign-off on provider terms, proofing requirements per grade, consent handling and evidence-reuse windows *(counsel)*.
10. Agent-economy parameters (schemas, scope vocabulary, mint and lease schedule, cut points) and — launch-blocking for the agent layer, not for the identity-only launch — the **accountability-escrow disclosure policy**: threshold, quorum, evidence schema and audit trail for unmasking the human behind an agent.

14.2 Decisions deliberately open

Custodial or facilitated payment addresses on the classical seller rail — proposed out of scope for the first version and reopened only if a material facilitator asks *(counsel)*; the KYC provider arrangements of item 9 *(counsel)*; offer sequencing, subscription currencies and the treatment of unsold tokens *(counsel)*; the instrument characterisation of the arbiter rung *(counsel)*; who posts the claim bond for facilitated x402 sellers; the lawful basis for delegated consent and the consent and anti-coercion texts per jurisdiction *(counsel)*; and EU AI Act conformity of disclosure receipts and cross-border liability *(counsel)*.

14.3 Risks, stated plainly

- **Competitive timing.** By April 2026 every major payment network had shipped or unveiled a transparent know-your-agent primitive; MintID will not win the low-assurance mass segment and does not try to. Its defensible niche — privacy-preserving, funded, regulated-grade backing — is validated as unserved but narrow; the consequence is sequencing, not repositioning.
- **Cold start.** Two-sided, addressed by the first implementer and ZadQ, the anchor product; success criteria are defined in advance and measured.
- **Cryptographic delivery.** The proof core is the critical unbuilt component; the self-revocation proof is the highest-risk consensus-path item and stays disabled until audited. A slip moves the zero-knowledge dependencies, not the identity-only launch — but the identity-only launch still needs the credential profile.
- **Regulatory.** MiCA execution, the Foundation’s board and statutes, GDPR and consent texts, AI Act conformity, product-liability posture and the unmasking governance are subject to counsel; the design chose the strictest

available reading at each point (Title II in full, the AEC invariant, no anonymity-enhanced reserve) to keep the characterisation uncontestable.

- **Economic.** The token investment case rests on bands whose points the simulation has not yet fixed; adverse adoption scenarios show a thin protocol P&L and must be presented, not averaged away; the raise floor is invariant, and re-staging is the committed answer to a weak offer.
- **Related-party concentration.** Development, the first issuer, the reference SDK, ZadQ and pre-genesis funding sit in one group; ZadQ's private deployment is operated by the same group and is therefore reported as a pilot and excluded from the traction criterion. The answer is disclosure and equal on-chain terms, not concealment, and the independent economic review — still pending — remains the only arm's-length opinion.
- **Delegated trust and fail-closed operation.** A relying party trusts issuers it does not choose, and the fail-closed rule means that a silent issuer takes its credentials offline within 45 seconds; both are the price of the guarantee, and both are stated to integrators without softening.

GLOSSARY

Glossary

Accepted issuer — a council-admitted, bonded KYC organisation whose credentials the protocol accepts. **Accountability escrow** — the issuer-private record linking an agent to its human; disclosed only by due process (distinct from the *financial* escrow). **AEC invariant** — the token never gains anonymity-enhanced transfer modes. **Agent credential** — a credential derived by an issuer from a principal's credential, carrying a backing assertion and a delegated scope. **Anchor product** — ZadQ, the x402 seller-verification service that is the first product built on MintID; operated by the first implementer, disclosed as a related-party position. **Anchor pilot deployment** — the private, self-hosted MintID installation on which ZadQ's first attestations run; not the network, not a launch, excluded from the traction criterion. **Anonymous credential** — a credential with a blinded issuer signature, a holder secret, selective disclosure and predicate proofs. **Assurance grade** — the private A1–A4 grade an issuer assigns. **Authenticated read** — a query of finalised chain state whose answer is proven against a light-client-verified header. **Authority state** — the closed set of states an issuer or verifier can be in: pending, active, enhanced monitoring, suspended, revoked. **Backing** — the fact that an accepted issuer stands behind a credential at a given grade; provable, unlike identity. **Band / point / prior** — a ratified range; the final value inside it, fixed by the simulation; an argued working point that the simulation validates. **Commercial rail / dispute rail** — the two firewalled disclosure paths (consent-gated vs due-process-gated). **Compliance council** — the threshold-vote operational body for admission, bonds and treasury co-signing. **Consensus firewall** — the rule that off-chain service revenue never influences consensus rewards. **Disclosure bid / dispute bond** — a non-slashable payment for consent vs slashable adjudicated stake; neither is ever a lawful basis for disclosure. **Emergency status key** — an issuer's separately scoped key that alone can publish an immediate root. **First implementer** — the group operating the first issuer, verifier, reference implementations and anchor product on undifferentiated terms. **Foundation share** — the earmarked, never-distributed portion of protocol cuts. **Freshness band** — the issuer-held band (`F1`–`F4`) saying how recently a holder's evidence was verified; a second dimension next to the grade, published as a band and never as a date. **Holder** — a person or principal controlling credential secrets. **Holder secret** — the never-disclosed secret that binds a credential to its holder. **KYA** — know-your-agent. **L0 / L1 / L2** — the money-before-identity cascade: escrow payout without identity; minimal fraud signal; due-process unmasking. **Nullifier** — the one-time value a self-revocation adds to the issuer's revocation set. **Operator floor program** — foundation-funded service agreements paying validators for verifiable service. **Pairwise pseudonym** — an agent's per-verifier identifier, stable for that verifier and unlinkable across verifiers. **Presentation** — a ten-second, fully bound zero-knowledge proof of requested predicates. **Principal** — the human behind an agent. **Reward pool** — the part of the maximum supply not minted at genesis, usable only for validator emission. **Session reservation** — a short-lived pledge of escrow capital to one session, under value conservation. **Standing read** — a verifier query answering whether a conventional seller attestation is active, suspended or revoked, from proven chain state; a classical read, not a presentation. **Status root** — an issuer's signed 30-second commitment to its private status registry. **Stichting** — the Dutch foundation form of the MintID Foundation. **Verifier** — a KYC-approved, on-chain-registered relying party bound to exact origins. **Witness** — the holder's private proof of a credential's status against the issuer's root.

REFERENCES

References

MintID specification corpus (frozen; available on request). The core specification of the sovereign privacy-preserving identity chain; the agent-identity and consensual-disclosure specifications; the Genesis Allocation and Transparency Policy; the Token Sale and MiCA Title II Plan; the economic-simulation scope; decision records and improvement proposals; and the MintID Issuer, Verifier & SDK, Holder and Validator manuals.

Standards and frameworks. Cosmos SDK; CometBFT; ICS-23 store proofs; OpenID for Verifiable Credential Issuance; OpenID for Verifiable Presentations; OpenID4VC High Assurance Interoperability Profile (SD-JWT VC / ISO mdoc — used as security guidance only); OpenID Connect for Identity Assurance; Certificate Transparency Merkle trees (RFC 6962); Ed25519 (RFC 8032); Hyperledger AnonCreds; W3C Verifiable Credentials Data Model (referenced, not adopted as the primary format); EU Digital Identity Wallet Architecture and Reference Framework; Regulation (EU) 2024/1183 (eIDAS 2.0), Art. 5f; Regulation (EU) 2023/1114 (MiCA), Title II, Arts. 3(1)(33), 7, 15, 76(3); Regulation (EU) 2024/1624 (AMLR); Regulation (EU) 2016/679 (GDPR), Art. 7(3); ISO/IEC 29115 (entity authentication assurance framework); ETSI TS 119 461 (identity proofing); Regulation (EU) 2024/1689 (AI Act); NIST SP 800-63 (IAL/AAL); x402 (Linux Foundation); Web Bot Auth and HTTP Message Signatures.

Market and comparative sources. OpenID Foundation, *Identity Management for Agentic AI* (South et al., Oct 2025); IBM Think, *The Practitioner's Guide to Non-Human Identities* (Feb 2026, Omdia 50:1 ratio); IDC (Villars, Dec 2025, more than one billion active agents by 2029); Gartner (agentic enterprise applications); KYA / KYAPay protocol documentation (Skyfire); Concordium Verified Agents; ERC-8004; Visa Trusted Agent Protocol; Mastercard Agent Pay; Google AP2; cheqd; Privado ID. All market figures are dated claims, refreshed before any governance decision depends on them; figures without a primary source are not used.

APPENDIX A

Core requirements (R1–R28)

ID	REQUIREMENT	ID	REQUIREMENT
R1	Sovereign, secure proof-of-stake chain with deterministic finality	R15	Online verification against the latest finalised state
R2	No raw KYC on-chain	R16	Native-token issuer bonds, slashable by an evidence-committed council process
R3	User custody of credentials and secrets	R17	Hard-capped monetary policy with a declining schedule and burns
R4	Multiple credentials, no public master identifier	R18	Privacy-preserving verifier workflow; presentations never on-chain
R5	Accepted (council-approved) issuer model	R19	Bounded state growth — issuers, verifiers, validators; never holders
R6	Private assurance grades A1–A4, threshold proofs	R20	Security-first release: audits, adversarial testnet, bounty, incident plan
R7	One-time issuance; renewal creates a new credential	R21	Selective disclosure and predicates *(should)*
R8	Root-based status, no per-user state	R22	Multi-claim ten-second sessions *(should)*
R9	30-second status freshness, fail closed	R23	Holder-side trusted verifier domains *(should)*
R10	Immediate issuer emergency revocation via an emergency key	R24	Verifier domain-control revalidation, no wildcards *(should)*
R11	Holder self-revocation revealing neither identity nor credential identifier	R25	Independent evidence commitments for council decisions *(should)*
R12	Recovery only by KYC re-verification	R26	Optional paid services, firewalled from consensus rewards *(should)*
R13	KYC-approved verifiers bound to domains and keys	R27	State pruning and snapshots *(should)*
R14	Ten-second, fully bound, perishable presentation	R28	First-party KYC integration behind a provider-neutral interface *(should)*

The agent-layer and disclosure-layer requirements are summarised in §7.

APPENDIX B

Ratified values, bands and priors at a glance

VALUE	STATUS	FIGURE
Maximum supply	**Frozen**	108,000,000, exact
AEC invariant; no post-cap mint; burns never re-mint	**Frozen invariants**	—
Genesis allocation — development / sale / reward pool	Bands 15 / 20–25 / ≥ 60 %	Prior 15 / 25 / 60 %
Emission curve	Rule: cumulative emission equals the reward pool; strictly declining	Prior 3 % of the cap in year one, ×0.95 per year
Flow-A protocol cut	Band 8–16 %	Prior 12 %
A/C cut split — burned / validator / foundation	Bands 20–40 / 35–55 / 15–30 %	Prior 30 / 45 / 25 %
Genesis active set / voting-power cap / unbonding	Priors	60 (⟶ 100 by governance) · 5 % · 21 days
Launch-gate thresholds	Priors	≥ 20 % externally bonded · ≥ 25 operators · Nakamoto ≥ 7
Raise floor	**Ratified invariant**	$4.0 M net of statutory withdrawals
Sale price mechanism	Ratified	A single fixed price, no tiers — the figure appears only in the Title II white paper
Vesting / lock-up / no-vote rule	Policy	Four years continuous, one-year cliff · 12 months · applies while vesting or locked
Status freshness	Governed parameters	30 s heartbeat · 45 s maximum age · fail closed
Presentation lifetime	Constant	10 s; single-use challenge
Council	Recommended	5-of-9 threshold; custodian co-signature for reserve movements
Foundation	Decided	Netherlands Stichting (Amsterdam), dormant until the public-offer route is chosen (ADR-0016); competent authority AFM
Assurance vocabulary	Decided (August 2026)	A1–A4 anchored on eIDAS low / substantial / substantial + EDD / high, squared with ISO/IEC 29115; freshness bands `F1` – `F4` as a second dimension
Reserve set	Ratified	BTC (with haircut discipline) plus fiat and stablecoin instruments; no anonymity-enhanced coin

MintID Protocol Whitepaper v1.0, 16 September 2026 (supersedes the v0.1 draft of 15 August 2026) · prepared by the MintID project, a CONECTIA-group initiative (related parties: §11.4) · non-normative; the specification corpus governs · not a MiCA crypto-asset white paper and not an offer,

solicitation or marketing of a crypto-asset · not legal, tax or investment advice.