MINTID · CONCEPT AND MARKET PAPER

Verifiable Human-Backed Agent Identity — Concept and Market Paper

A person verifies their identity once. Every software agent that person deploys can then prove, to anyone, that a verified and accountable human stands behind it, without revealing who that human is.

PUBLISHED BY

MintID

Governance corpus (master/) · developed and operated by Conectia, a disclosed related party

VERSION	DATE	STATUS
v3.9	3 November 2026	Concept and market paper

Contents

01	Summary	4
02	The problem	5
03	The use cases, in plain words	6
04	What MintID is	8
05	Agents backed by people	10
06	Two ways identity can leave the system	12
07	Why this design and not a bearer-token service	14
08	Market and go-to-market	15
09	Economics in plain words	17
10	Governance, legal posture and related parties	18
11	Where the project stands and what comes next	20
12	Risks	21
13	Appendix — where each claim is specified	22

STATUS: Concept and market paper (non-normative). The specifications (`SPEC-1`, `SPEC-2`, `SPEC-3`) and the policies of this corpus govern on every point; this paper explains them.

VERSION: v3.9 (2026-11-03). Change history: `git log --follow -- master/Agent-Identity-Concept-and-Market-Paper.md`; decisions: `docs/adr/`.

AUDIENCE: investors and financing counterparts, partner executives, the management of KYC organisations, product owners of services that accept agents as customers, policy and compliance readers.

ONE-LINE THESIS: A person verifies their identity once. Every software agent that person deploys can then prove, to anyone, that a verified and accountable human stands behind it, without revealing who that human is. Money settles most disputes before a name is ever spoken, and a name is disclosed only with the person's priced consent or under due process.

KYC-backed identity and accountability for the agentic era. *Zero-knowledge identity layer for agents.*

01

Summary

MintID is a purpose-built identity network. Approved identity-verification organisations (issuers) verify people and hand them a credential that the person keeps. With that credential, a person, or an agent acting for that person, can prove facts such as “verified at grade A3 or better”, “over 18”, “credential still valid”, or “authorised to pay this seller up to this amount” to a registered service, in ten seconds, without disclosing anything else. The network holds the trust roots the service relies on. It never holds personal data, credentials, or a record of who proved what to whom.

The design answers a problem the agent economy created. Identity systems assume a human at the keyboard who can log in, consent and be held responsible. Agents act on their own, in fleets, across organisations and at machine speed. Making an agent accountable has meant attaching a human’s real identity to it; keeping the human private has meant having no one to hold responsible. MintID removes that trade-off: an agent proves it is backed by a verified, liable person, and the person stays private.

The model has four layers. First, human identity verification with privacy: verification once, proofs many times, no tracking. Second, agents backed by people: each agent gets its own credential derived from its principal’s, carrying a delegated scope; no two agents of one person can be linked; a holder can kill a credential without the issuer’s help and without revealing which one. Third, accountability when something goes wrong: a harmed counterparty files a claim against the agent’s pseudonym; the person can settle it privately, an independent arbiter sets an offer if they disagree, and only a substantiated claim left unpaid, or one the arbiter sends to court when the claimant rejects the offer, leads to the person’s identity, under due process. MintID holds no money at any step. Fourth, consented disclosure: a service that needs to know its customer offers a price, the person accepts or refuses, and a refusal costs nothing.

Where the project stands: the single-node production chain has been in production since 2026-09-20, operated by Conectia for ZadQ, the first product built on the protocol. The chain, the issuer and verifier services, the anonymous-credential core, the agent credential and the holder’s kill switch run there; a public testnet with a KYC sandbox is the one deployment open to third parties for testing; the code and its packages are public under Apache-2.0. The consented commercial disclosure rail is built, switched off and fail-closed, and its activation waits for counsel. The dispute rail is specified; the protocol has no financial layer and holds no funds. No public network exists yet, no token has value, and no token or crypto-asset is for sale. Independent audits are scheduled milestones whose scope and findings will be published; no component is described as independently audited before its audit has closed.

Conectia and ZadQ are related parties of MintID and are disclosed as such throughout this paper. The production chain is a related-party pilot; it is neither the MintID network nor a launch, and nothing measured on it counts as independent adoption.

02

The problem

Every identity system of the last twenty years was built for a human at the keyboard. A person logs in, sees a consent screen, clicks, and can later be asked what they did. Software agents break each of those assumptions. They act without a person present, they run in fleets of hundreds or thousands under one owner, they cross organisational boundaries in a single task, and they act at a speed no consent screen can follow. The OpenID Foundation's paper on identity for agentic AI (South et al., October 2025) states the consequence in four words: with autonomy comes accountability.

Current practice bolts agents onto the plumbing built for machines: service accounts, API keys, access tokens. That plumbing fails on four fronts at once.

- **Accountability.** Nothing in a service account proves that a liable party stands behind the agent. A counterparty can see that an agent is running; it cannot see whether anyone answers for it.
- **Blast radius.** When an agent is given its owner's credentials, a stolen agent credential is a stolen human credential. One compromised agent exposes the person's whole identity.
- **Sybil resistance.** Agents register with services anonymously and in bulk. The same paper describes the result as a complete lack of a paper trail: an actor that is blocked re-creates itself under a new name.
- **Privacy.** The only way to make an agent accountable under current tools is to attach the human's real identity to it. Accountability and privacy are treated as opposites.

The Agentic AI Foundation (AAIF) taxonomy defines accountability as "the processes and mechanisms ensuring that AI agents operate within defined ethical, legal, and operational boundaries with traceability for their actions and decisions"; MintID's accountability differs in three ways: no record of what an agent presents is kept beyond the few items a dispute needs, held by the verifier for a fixed claim window and then deleted, a person is identified only with their own authorisation or under due process, and in a dispute identity comes last, after a private settlement and an arbiter's offer (specified, not in production; §6.2).

The scale is large. Non-human identities outnumber human ones by roughly fifty to one in enterprise environments (Omdia, December 2024, cited by IBM Think, February 2026). Industry analysts expect more than a billion active agents by 2029, up from about 28.6 million in 2025 (IDC, December 2025), and about 40% of enterprise applications to embed agents by 2026, from under 5% in 2025 (Gartner). These are analyst projections, not measurements, and are labelled as such. The direction is not in doubt: the population that needs an accountable identity is growing faster than any per-person verification model can serve.

A further consequence is worth stating plainly because it is often only asserted. A network of agents cooperating under uncertainty stays usable only if participants can tell who stands behind a counterparty. An agent cannot judge from a message alone whether the party behind it is genuine. The question has to move from the message to the source, and it can only be answered by a shared, verifiable layer outside the agents themselves. MintID is that layer.

03

The use cases, in plain words

Eight things the layer does, for a first-time reader, each with where it stands. The use-case documents of this corpus carry the precise version and govern on conflict.

A person verifies once and proves many times. A KYC issuer verifies the person and gives them a credential they hold themselves. From then on they answer a service's question ("over 18", "verified at grade A2 or better") with a ten-second proof the service can check; the service learns the answer, never the person. No new verification per service, no tracking across services. *In production on the single-node production chain.* (`Use-Case-Human-Identity-Verification.md`)

Every agent carries a verified person behind it, and a scope that person fixed. The person mints one credential per agent. The agent proves, to any relying party, that an accountable human backs it and that the action is inside its scope, without saying who the human is. One verification covers a whole fleet; an operator's agents cannot be linked to each other. *In production; the bounded agent (an agent minted with a preset scope) ships in the issuer software, with the scope bundles the operator publishes.* (`Use-Case-Human-Backed-Agent-Credential.md`)

The kill switch. The person, or the operator, switches a credential or an agent off without asking the issuer and without telling the network which credential died. Every service refuses it within about four minutes of the network recording the switch-off (245 seconds on the single-node production chain; 110 seconds at the protocol's reference values). It is not instant: until the next status commitment carries the revocation, a presentation can still be accepted. An issuer's own revocation is refused everywhere within about three minutes there (185 seconds), or at once in an emergency. *In production since 24 September 2026; its independent audit is a scheduled milestone.* (`Use-Case-Holder-Self-Revocation.md`)

A buying agent a seller can let in. The agent attaches a ten-second proof to its paid request: human-backed, within scope, not revoked. The seller decides on the proof instead of blocking everything that is not a browser. *In production on the single-node production chain, end to end (related-party pilot).* (`Use-Case-x402-Buyer-Agent-Identity.md`)

A seller that charges per request proves it is an accountable business. ZadQ, the first product built on MintID and a disclosed related party, lets a seller on the x402 payment rail show every buyer that an accountable, identity-verified person stands behind the endpoint; a business is represented by the person who acts for it, and the verification of the business itself is deferred. *In production for ZadQ on the classical seller rail (issuer-signed attestations); the zero-knowledge seller rail follows on engineering readiness.* (`Use-Case-x402-Seller-Verification.md`)

Identification, not surveillance. When a seller legitimately needs to know its customer, it asks for one graduated item of information and may offer to pay for it; the person's consent policy decides, and refusal, timeout or error disclose nothing. A payment is never the lawful basis, and knowing once confers no right to recognise again. *Built, switched off and fail-closed on both sides; activation waits for counsel.* (`Use-Case-Consented-Commercial-Disclosure.md`)

A nominated, legally obligated counterparty learns what the law needs, and nothing else. A person can mandate in advance that a named obligated party receives a bounded set of attributes, capped and expiring; everyone else receives the proof, not the person. *Built in the issuer software and switched off behind a second switch, separate from the commercial rail's own; the obligation-class schedule, seeded in the software, reaches the chain at the next regeneration, and the channel that delivers a release to the counterparty is specified; activation waits for counsel.* (`Use-Case-Mandated-Identification.md`)

Identity last in a dispute. When an agent misorders or defrauds, the wronged party does not need the operator's name; it needs to be made whole. A claim bound to the interaction reaches the person behind the agent through the issuer, without naming them, and the person can settle it directly. If they do not, an independent arbiter decides whether the claim is substantiated and sets an amount; only a substantiated claim left unpaid, or one the arbiter sends to court when the claimant rejects the offer, releases the person's identity, under due process. MintID holds no funds at any step. *Specified, not built; the dispute rail activates on a published organic-demand criterion.* (`Use-Case-Dispute-Cascade.md`)

04

What MintID is

MintID is a sovereign identity chain: its own network, its own validators and its own rules, built with mature blockchain infrastructure rather than a new consensus engine. Its purpose is narrow. It lets approved identity-verification organisations issue credentials to people, and it publishes the trust roots that let anyone check those credentials are current. It is not a place where identities are stored and it is not a log of identity checks.

Three roles. An **issuer** is an organisation admitted by the MintID Foundation under a published policy (by the compliance council, its organ, once that council is seated) to perform identity verification (KYC) and issue credentials. A **holder** is a person who completed verification with an issuer and holds the credential on their own device, under their own keys. A **verifier** is a service, registered on the network with its exact web address and admitted by the Foundation, with a responsible person behind it verified by an accepted issuer, that asks a holder to prove something. Issuers post a bond in the network's token that can be forfeited for proven misconduct, and they are accountable to the Foundation and, once it is seated, to the council.

What is on the chain. The register of approved issuers with their public keys and status; a small signed commitment from each issuer, refreshed every thirty seconds, that summarises the current status of everything that issuer has issued; the register of approved verifiers with their web addresses and keys; the records of council decisions; and the token's economics. The size of this state grows with the number of issuers, verifiers and validators. It does not grow with the number of people, credentials or checks.

What is never on the chain. No name, address, date of birth, document, biometric or any other personal data. No credential and no credential serial number. No record of a proof: a presentation travels directly from the holder to the verifier and is never broadcast, indexed or stored on the network. No per-person entry of any kind. These are hard rules of the specification, not configuration.

Thirty-second freshness. Each issuer publishes a signed status commitment at least every thirty seconds. A verifier that sees an issuer fall silent treats every credential from that issuer as unusable until the issuer resumes. Revocation therefore reaches every verifier within a bounded time, under a minute at the protocol's reference values and about three minutes on the single-node production chain (185 seconds), and an issuer that fails or is compromised fails closed rather than open. An issuer also holds a separate emergency key with which it can revoke immediately, on its own, in a way the council can review afterwards.

The ten-second proof. When a verifier wants to check something, it sends a signed challenge that names itself, its exact web address, the facts it wants proven, and the current state of the chain, valid for ten seconds. The holder's software builds one proof that answers exactly those facts and nothing else, bound to that challenge. The verifier checks the chain again at the moment of decision, records a minimal decision, and discards the proof. A proof cannot be replayed, redirected to another service, or reused after ten seconds. The verifier learns the issuer's identity and the answers to its questions. It does not learn the person's grade, expiry date, credential identifier or any attribute it did not ask for.

Grades. Issuers assign one of four assurance grades, from basic identity binding to enhanced due diligence with periodic review. The grade vocabulary is aligned with the European eIDAS assurance levels and ISO/IEC 29115. The exact grade stays private; a holder proves "at least the grade you asked for".

Recovery and renewal. There is no social recovery and no password reset. A person who loses their keys verifies again and receives a new credential that cannot be linked to the old one. Renewal works the same way. The absence of a persistent person-level identifier is deliberate: it is what makes tracking impossible by construction.

05

Agents backed by people

The same machinery, applied once more, turns a person's verification into an accountable identity for every agent that person deploys.

One verification, many agents. A person who holds an identity credential can ask their issuer to mint an agent credential. The person proves, to their own issuer and in zero knowledge, that they hold a valid credential of sufficient grade. The issuer mints a fresh credential for the agent that carries three things: the fact that a verified human stands behind it, the grade of that human's verification, and a delegated scope. The issuer keeps the link between the agent and the person in its own private records, under the same protection as the verification evidence itself. Nothing about the agent is written to the chain: to the network, an agent credential is simply one more credential under its issuer's status commitment.

What an agent proves. To a counterparty, the agent proves that it is operated by a verified, accountable person at a grade of at least G, that it is authorised for this action with this counterparty in this category and region up to this value, that its credential is active and unexpired, and, for the counterparty's own records, a pseudonym that is stable for that counterparty only. It proves all of this in one ten-second presentation. It reveals neither who the person is nor which other agents that person runs.

Delegated scope. The scope is fixed by the issuer at mint time along five axes: what the agent may do, with whom, in which category, where, and within which validity window and up to which value per transaction. Each axis can be proven separately without revealing the others. A cumulative limit across counterparties is not something the credential proves: within one counterparty it is that counterparty's own session accounting (`SPEC-2` R-A5.1). An agent cannot widen its own scope; a new scope means a new credential. A counterparty that sees no statement on an axis it cares about refuses by default.

No two agents are linkable. Two agents of the same person are unlinkable to the network, to any verifier, and to two verifiers comparing notes. A counterparty that recognises an agent recognises that agent alone, for the life of its credential, and only under a pseudonym that no other counterparty shares.

The kill switch. A holder can revoke a credential without the issuer's cooperation and without telling the network, or anyone watching it, who they are or which credential died. The holder submits a one-time token with a proof through a privacy relay that pays the transaction fee on the holder's behalf and keeps no record of where the request came from. The network verifies the proof and records the token; a consensus rule then forces the issuer to reflect the revocation in its next status commitment within a fixed grace, on pain of every credential from that issuer failing closed. The effect is bounded by consensus: every verifier refuses the credential no later than 110 seconds after the network records the token at the protocol's reference values, and 245 seconds on the single-node production chain, whose status commitments stay accepted for up to 180 seconds. It is not instant, since until the next status commitment carries the revocation a presentation can still be accepted; and no verifier depends on being told, because every verifier reads the same status commitment. When the human's own credential is revoked this way, the issuer revokes that person's agents as well, spread over several status commitments so that an observer cannot tell that one person's fleet went dark.

The payments use case. The first use of the agent layer is in x402, an HTTP-native payment rail in which an agent pays a seller per request in stablecoins. Two sides of that market need an accountability anchor. A **seller** listed on a discovery surface proves that an accountable, verified person or organisation stands behind its payment address and its service, without revealing who; explorers and buying agents consume that signal, and a registered verifier re-checks sellers on a published schedule and publishes its own signed verdict. A **buyer agent** proves to the seller, or to

the payment facilitator acting as the seller's verifier, that it is human-backed, at a sufficient grade, and delegated to pay this seller within its limits, once per session and inside the same request that carries the payment. The payment itself stays outside MintID: MintID holds no payment key, chooses no payer address and settles nothing, and whether a payment links its payer across sellers depends on the wallet and the payment scheme. Both sides are described in the use-case documents of this corpus.

06

Two ways identity can leave the system

Most of the time no identity leaves the system at all: counterparties act on proofs. Two situations require more, and they are handled by two separate mechanisms that share nothing.

6.1 Consented commercial disclosure

A service that wants a relationship with a customer, to open an account, bill a subscription, or meet its own regulatory duty to know its customer, needs identity detail that a proof does not give. On the commercial rail the service makes an offer: it names the minimum tier of detail it needs, from a pseudonym it can recognise, to a single contact attribute, to full verified identity, and it names a price. The person's issuer checks the offer against the rules the person set when the credential was minted. Small, routine offers within the person's policy are accepted automatically; anything above it is pushed to the person's device for a live decision; no answer means refusal. On acceptance the issuer releases exactly the requested field and nothing more, the price is paid to the person, and a content-free receipt is committed to the chain so that abuse can be audited later. On refusal nothing happens and nothing is owed.

Three rules make this a privacy product rather than a loophole. Refusal is free and safe: a service must always offer a way to complete the core transaction without disclosure, and disclosure may only unlock extras. Full identity is never released without a live human decision, however long the agent has been running on its own. And a disclosure identifies one agent at one moment: it gives the service no standing right to recognise or track the person afterwards. That is a rule on the service, which holds what it received as its own record under its own data-protection duties, and a non-provision by MintID, which gives the service no handle that survives renewal. A service that wants to know the same person again after the credential renews pays again, with consent again.

The lawful basis for a commercial disclosure is the person's own consent, given for a specific purpose at a specific moment. Payment is never the lawful basis.

6.2 The dispute path: identity last

When something goes wrong, the party who was harmed cannot be asked to wait for the wrongdoer's consent; a wrongdoer would always refuse. The dispute rail therefore never asks the person. It follows due process, in an order that reaches for identity last.

- **First, settlement.** A harmed counterparty files a claim bound to the specific interaction and to the agent's pseudonym, with a proof of harm, and advances the arbiter's fee. Every agent's presentation carries an encrypted reference to the agent that neither its issuer nor the MintID Foundation can open alone. Once the claim is recorded on the chain, the Foundation adds its part, and the issuer finds its agent, whether or not the agent cooperates, and notifies the person behind it without revealing them to anyone. The person can pay the claimant directly, by their own means; the claim then closes and no identity is revealed. If the person contests or does not answer, an independent arbiter chosen at random from a registered pool, who sees only the pseudonym and the proof of harm, never a name, decides whether the claim is substantiated and sets a settlement amount. A claim found unsubstantiated closes, and a frivolous one costs the claimant its fee. Most disputes are expected to end here.
- **Second, a minimal signal.** If the person neither pays nor contests a substantiated offer, the issuer releases a minimal verified signal, for instance "a real verified person, grade A2, in this region, with this many prior

substantiated disputes", enough to price the risk and not enough to identify anyone; the issuer revokes the agent where the arbiter so decides.

- **Last, identity.** If the person does not pay the offer, or if the claimant rejects it and the arbiter decides the case belongs in court, the issuer, the only party that holds the link, releases the person's identity to the claimant under the arbiter's certified finding, for the claimant to pursue in court; the compliance council supervises and audits the release after the fact and never originates it. The release is never purchasable, and a refusal to pay a claim the arbiter has not found substantiated releases nothing.

No money passes through the protocol. MintID, the Foundation, the operator and the issuers hold no funds and no key that moves them; a settlement is a payment from the person to the claimant, by the person's own means. The person behind every agent is the recourse.

Status. Both mechanisms are specified in full; neither is active. The commercial rail is built and switched off, fail-closed on both sides; the dispute rail is specified and not built. Both activate on a public network only when a published organic-demand criterion is met, with the counsel-reviewed governance texts on consent, anti-coercion and the identity-release threshold in place.

07

Why this design and not a bearer-token service

The nearest alternative in operation is a centralised “know your agent” service: one operator verifies people and organisations and issues a signed token that an agent carries to sellers. KYA/KYAPay, operated by Skyfire, is the reference example and it operates with ordinary web infrastructure, because it gives the seller what a seller often wants: the customer’s verified identity, immediately. The token carries the person’s verified email address to every seller that receives it, together with the further verified fields the seller requires or the buyer chooses to share, such as name and date of birth (Skyfire developer documentation, KYA Token and Identity Fields, July 2026); it is signed by a single trusted issuer and may be used multiple times until it expires (Skyfire developer documentation, Common Token Claims, July 2026). The validation rules of the KYAPay profile check the token’s expiry and define no revocation check (KYAPay profile, IETF Internet-Draft draft-skyfire-kyapayprofile-02, June 2026, §4.1.2). There is no unlinkability between an agent and its siblings and no way for a seller to obtain recourse without holding the identity.

MintID serves the same commercial need in a different order. By default a seller learns that an accountable, verified person stands behind the agent, and, if harmed, can be settled with privately by that person, without learning who the person is. When the seller needs to know the customer, it asks and pays, and the person decides. When the seller is harmed, a private settlement comes first and identity comes last, under due process. Trust does not rest on one operator: any approved issuer can issue, and every proof is checked against public trust roots that the network’s validators maintain. A reusable identity token is itself a tracking surface; MintID has none. Even a paid reveal leaves the person unlinkable across their other agents and across renewal.

Three other approaches define the field. **Concordium’s Verified Agents** put an agent registry and protocol-level identity on a sovereign chain and have live merchants; its identity layer is built for verification: the verified party stays private to ordinary observers and is identifiable under legal process; MintID is built for accountability with privacy. **ERC-8004** gives Ethereum an open standard for public agent records; it shows that agent identity will be a shared layer, and its records are public by design while the owner behind a record can be fully anonymous. **Visa Trusted Agent Protocol and Mastercard Agent Pay** run agent trust on the card networks’ own settlement; they legitimise the category and will serve the low-assurance, card-adjacent segment. MintID’s segment is the one those rails structurally cannot serve: privacy-preserving, KYC-grade human backing with crypto-native funded recourse and no personal data spilled to the counterparty.

The timing risk is real and is stated as such. Every major payment network shipped or announced an agent-identity primitive by April 2026; Skyfire demonstrated its service with Visa’s agent commerce programme in December 2025 and signed an enterprise distribution partnership. If the industry standardises on transparent tokens before a privacy-preserving alternative is usable, that alternative becomes an upgrade no one integrates. That is why the identity-only layer runs in production first and the commercial rail is decoupled from every financial-layer milestone.

08

Market and go-to-market

The served segment. The first market is the x402 ecosystem: sellers that charge per request in stablecoins, the buyer agents that pay them, the marketplaces that list them and the payment facilitators that settle for them. Its discovery layer lists any endpoint after one settled payment, with no identity or vetting anywhere and with trust signals that are cheap to inflate. Reported activity in late April 2026 was on the order of tens of thousands of active agents and hundreds of millions of transactions, with an average transaction of twenty to thirty cents (industry reports; Chainalysis put annualised volume near $600 million by March 2026). Those figures are external and dated and are refreshed before any decision relies on them. The question that grows with value at risk and with incoming regulation, whether an accountable person stands behind this endpoint or this agent, provable without revealing who, is the MintID product statement. The served-segment declaration of the anchor product (ADR-0011) is a statement about its seller line and about the traction criterion measured on it; every other segment, the accountable principal of bounded agents at any volume and the business that accepts accountable agents among them, is declared, in its own terms, by the operator of the commercial layer that serves it (ZadQ today). One rule holds on every surface, MintID's and any operator's: MintID is never sold as a replacement for a statutory KYC or anti-money-laundering check; the saving that may be claimed is on the checks no law requires, where one verification backs every agent a person deploys (ADR-0033).

The anchor product. ZadQ operates the first product on MintID: an x402 seller-verification service. A seller completes identity verification at the issuer boundary and receives an attestation (an organisation is represented by the person who acts for it; the verification of the business itself is a later phase); explorers and buying agents read the standing of that attestation; a registered verifier re-checks sellers on a published cadence and publishes its own signed verdict. ZadQ's client uses the classical form of that rail, in which the issuer signs the attestation directly; the zero-knowledge form is mounted on the same chain, ZadQ's client-side pieces for it are built, and the move is ZadQ's to record. The classical form is described as classical and never as zero-knowledge.

The two-sided cold start. An identity network has no issuers without verifiers and no verifiers without credentials. MintID answers with a first implementer: Conectia operates the network's first accepted issuer and first registered verifier on the same terms as any other party will: the same admission path as any other party (a decision of the MintID Foundation under its published policy, executed by the operator through a governance proposal voted by the chain's single validator until the compliance council is seated, and marked as a related-party decision), the same forfeitable bond and the same thirty-second status commitments, all of it checkable on the chain. The position carries no privilege and is disclosed as a related-party position. Adoption is measured only from independent parties: the traction criterion that gates the public route counts independent sellers, retention and independent verifiers, and treats Conectia's own deployment as context.

Sequence. The seller-side rail first, because it exists and has a customer. The buyer-side agent identity next, on the same chain, so that both sides of the x402 trust perimeter can be served. Then the dispute rail, which lets a harmed party reach the person behind an agent, settlement first and identity last, and the commercial disclosure rail, which lets a service know its customer on the customer's terms. An issuer runs the reference issuer software in its own infrastructure (the verification hand-off, the thirty-second status commitments, the revocation fold); its identity-verification provider sits behind the issuer's adapter and runs nothing of the protocol. Issuers are recruited on the argument that one recruited issuer brings its whole portfolio of verified customers, and that every verified person becomes many verified agents.

The relying-party route. Since October 2026 a business that accepts agents becomes a verifier itself, through ZadQ: an operator account under which the business's customers mint bounded agents, and the verifier software, self-

hosted or as a deployment operated by ZadQ under the business's own web address and keys. The business decides on every presentation itself; ZadQ never decides for it. A third-party verifier is admitted on the single-node production chain by the MintID Foundation under a published policy, once an accepted issuer has verified the responsible person behind it; the operator executes the decision, and admission requests reach the operator through ZadQ. What an operator sells is the enrolled agent per month or the operated deployment, never the presentation: verification itself stays free (§9). A fixed periodic seat that operators of the commercial layer would pay to Conectia, the chain's operator, is proposed and not in force.

The unit-economics argument. Digital identity verification is priced per check, at roughly twenty cents for a basic check and one and a half to four dollars per approved user for a bundled onboarding (Juniper Research, 2025, and vendor price lists; estimates). In a per-check model every agent of every person pays for verification again at every counterparty. On MintID one verification backs every agent that person deploys, at the marginal cost of minting and keeping an agent credential, and the protocol charges nothing to check a proof: a verifier that runs its own service pays nothing. The marginal cost of accountability per agent tends toward the cost of a mint, not the cost of a verification. The mint fee has a floor as well as a ceiling: low enough for agent economics, high enough that fabricating agents is never cheaper than passing verification. Where that corridor lies is a deliverable of the external economic simulation, not a figure this paper states.

09

Economics in plain words

MintID has a native token whose job is the security and governance of a public chain: validators lock it to earn the right to produce blocks and are penalised in it for misconduct, issuers lock it as a bond against fraud, and holders of it vote on the software and its parameters. The supply is capped at 108,000,000 tokens and can never be exceeded; validator rewards follow a published, declining schedule that ends when the reward pool is exhausted, and tokens removed by fees or penalties are gone for good.

If a public chain is launched, its initial allocation follows ratified bands: 15% to development, vested over four years with a one-year cliff; 20% to 25% to a public sale conducted entirely under the European MiCA regulation; and at least 60% to the validator reward pool. The exact points within those bands, the emission curve, the fee levels and the launch thresholds are outputs of an independent, external economic simulation, commissioned only if the public-offer route is chosen. Several allocation details are under review and are deliberately absent from this paper.

The protocol earns on the activity that scales with the agent population: a fee to mint an agent credential and a recurring fee while it lives, and a cut of consented disclosures, collected in aggregate so that no per-person record is created. The protocol charges nothing to verify: there is no protocol fee and no usage tax, a verifier that runs its own service pays nothing, and verification stays off the chain; that is a rule, not a pricing choice. Services that operators sell on top, such as operated verifier deployments, admission and onboarding, support, mirrors, archives, dashboards or paid verification APIs, are theirs to price under R26, outside the protocol, and never influence the chain’s rewards. Arbiter fees are paid by the parties to the arbiter, off chain. Every accepted issuer pays the MintID Foundation a flat annual admission fee, the same for every issuer, as a published condition of admission; it is not a chain fee.

No token with value exists. The production chain runs on a private token that is worthless by construction and will not be carried over to any public chain. No white paper under MiCA exists, no offer is open and no token or crypto-asset is for sale. The public testnet’s tokens are freely obtainable from its faucet, resettable and non-convertible.

10

Governance, legal posture and related parties

Two organs. Token-weighted governance owns the software: upgrades, consensus parameters and every versioned protocol parameter. The operational gates belong to the MintID Foundation: under a published policy it admits, suspends and revokes issuers and verifiers. On the dispute path it holds one part of the key that opens the encrypted reference to a disputed agent, and adds it, as a ministerial act, only for a claim recorded on the chain. A compliance council, a threshold-vote body with a recommended nine seats and a five-of-nine threshold, is the Foundation's organ once seated: it takes those gates on the Foundation's behalf, adjudicates bonds with public reason codes and evidence commitments, and co-signs protected treasury actions with an independent custodian. Neither the Foundation nor the council changes the software, and on the dispute path the council supervises identity releases after the fact; it does not originate them.

The MintID Foundation. The MintID Foundation holds the protocol's authority: it admits issuers and verifiers under a published policy. Conectia OÜ (trading as Conectia PRO) operates the single-node production chain under its mandate. An admission that meets the policy, the issuer admission fee included, is granted and one that does not is refused with a reason, so nothing is left to discretion, and every decision is published in the admissions log, with the decisions that concern a related party marked as such. The Foundation issues nothing, certifies nothing and verifies nobody; it holds no person's data and runs no node. The Foundation also performs the genesis of any public chain. Whether a public token offer is made is a separate decision, taken at Gate 1. If that route is chosen, independent board members, statutory locks and the council as an organ of the Foundation are in place before any white paper is signed; the Foundation then holds the treasury and the lock-up commitments and publishes the allocation statement whose hash is committed in the genesis file. As offeror, its notification authority under MiCA is the AFM.

Conectia. MintID is developed by Conectia OÜ (Estonia, registry code 16847069), trading as Conectia PRO, under the MintID name, and Conectia operates the single-node production chain under the Foundation's mandate until the protocol has traction. The mandate is unpaid and creates no claim in either direction; any economic relationship between Conectia and the Foundation starts only when the Foundation enters it, at arm's length and approved without the founder where the founder is on both sides. All code is published under the Apache-2.0 licence, so the Foundation will never need an assignment of intellectual property. The MintID trademark is to be registered in Conectia's name with a written commitment to assign it, or license it exclusively, to the Foundation. Conectia receives no advance, credit, reimbursement or compensation for its pre-genesis spending, which is its own research and development, disclosed as an internal budget of €400,000. Its return comes from ZadQ and from products built on the protocol, on the same published terms as anyone else's.

Related parties. The founder, Marc Molas, is the founder of MintID, the beneficial owner of Conectia and the founder-director of the Foundation; he is recused from every discretionary Foundation decision that concerns Conectia, and an admission under the published policy, which leaves nothing to discretion, is marked as a related-party decision in the public admissions log whenever it concerns Conectia or ZadQ. Conectia is the development company, the first accepted issuer and first registered verifier, the reference-implementation and premium-services operator, the operator of the production chain and the pre-genesis funder. ZadQ operates the anchor product and is a related party. Every one of these positions is disclosed in the corpus, in any offer document and in the transparency reports; this paper claims no independence it does not have. The only opinion that will be arm's length is the external economic review.

MiCA. If the token is ever offered, it is a Title II "other crypto-asset": a utility and network-security token, not an asset-referenced or e-money token, with no claim on any entity's revenue. Exactly one document will ever be a MiCA

crypto-asset white paper; this paper is not it and must not be presented as such. Until that white paper is notified, no token is marketed and no token or crypto-asset is for sale. Product and service marketing is unrestricted.

GDPR and eIDAS. Issuers are the data controllers of the verification evidence, which stays in their own systems under their own lawful basis and retention rules; the network never receives it. Verifiers keep a minimal decision record; for an agent's presentation they also keep, for a fixed claim window and then delete, what a dispute claim would need, including the encrypted reference to the agent that only its issuer and the Foundation together can open; nothing else of a presentation is kept. The commercial rail runs on the person's consent, the dispute rail on legal obligation and due process. The grade vocabulary aligns with eIDAS assurance levels, and the credential formats use the same issuance and presentation protocols as the European Digital Identity Wallet, so that MintID can be one of the verification stacks EU relying parties choose when their acceptance obligations begin.

11

Where the project stands and what comes next

In production on the single-node production chain (in production since 20 September 2026). The single-node production chain, `mintid-anchor`, runs in Spain, operated by Conectia for ZadQ. On it: the issuer and verifier registries with the issuer bond; thirty-second status commitments; the verifier registry with an admitted tier and a self-registration tier; the compliance-council and treasury modules (the council is built, not seated: admissions are decided by the MintID Foundation under its published policy and executed by the operator as governance proposals voted by the chain's single validator until it is); the anonymous-credential core with both the identity and the agent credential profiles; issuance through a certified external verification provider; the ten-second presentation with the agent predicates and the per-verifier pseudonym; the holder's kill switch with its privacy relay and the consensus rule that forces issuers to honour it; the payment-rail middleware and agent clients for x402; an issuer console, a verifier console and an onboarding portal, brought up on demand; the admission of third-party verifiers by the MintID Foundation under a published policy, with requests through ZadQ; and the end-to-end run of the agent channel on the live chain. ZadQ's sellers use the classical form of the seller rail. Beside it, the public testnet with its KYC sandbox is the one deployment open to third parties for testing, and an independent operator has run a published external test round on it.

Built, switched off. The commercial rail of consented disclosure, fail-closed on both sides, and mandated identification on it, behind a second switch separate from the rail's own; activation waits for counsel.

Specified, scheduled. The admission grant: the chain checking the Foundation's signed admission of every verifier, which replaces the operator's funding as the control. The dispute rail: the claim with its proof of harm, the private settlement, the arbiter market and the identity release under a certified finding. The operational hardening required before any unrelated third party depends on the chain.

The sequence. The first third-party verifier admitted on the single-node production chain. ZadQ's move to the zero-knowledge seller rail, whose client-side pieces are built. The adversarial campaign and the two audit engagements, the implementation audit of the proof core and the audit of the kill switch, with the same auditor in sequence; their scope is published before they start and their findings when they close. The dispute rail. A public adversarial testnet with independent validators, issuers and verifiers. Then Gate 1: the financing route is decided, between own capital, equity, public grants and a public token offer. If the offer route is chosen, the external economic simulation and a coordinated counsel engagement precede a MiCA white paper, the offer, and a genesis that is triggered by conditions rather than by a date: a published traction criterion measured on independent parties, an acceptable raise, and the simulation and counsel outputs delivered. The target is the second half of 2027, with 24 December 2027, the date on which eIDAS relying-party acceptance obligations take effect, as the reference. If the conditions are not met, the current mode continues, funded by Conectia, with no loss to anyone and nothing to reimburse. Dates and card-level status are in the roadmap.

12

Risks

- **Recourse without funds.** A principal may refuse to settle; the claimant then holds an identity and a case for court, not money. Mitigation: the arbiter's offer, the fraud signal and the revocation of the agent come first; the identity release under a certified finding follows; a relying party that needs funds at stake obtains them outside the protocol (a deposit or an insurer of its choosing).
- **A single operator in the endogenous phase.** The production chain has one validator, operated by a related party. Mitigation: the chain is described as a related-party pilot and never as the network; its security is key custody, and the keys are held in a managed key service; every regeneration rotates every key; a multi-node network with independent validators is the distributed target, and nothing measured on the single node counts toward traction.
- **Cryptographic implementation.** The credential core and the kill switch use mature, standardising primitives and no novel circuit, but their implementation audits remain to be completed. Mitigation: the audits are contracted milestones with published scope and findings; audit findings on the running system are treated as production incidents; no public wording claims an audit before it closes.
- **Regulatory.** MiCA, GDPR and eIDAS each touch the design. Mitigation: no token is offered or marketed before a notified white paper; the identity layer is designed so that the network is never a controller of personal data; counsel workstreams are scoped for each item marked as counsel-gated, and the consent and identity-release rules are activation-blocking governance items that cannot be skipped.
- **Adoption and timing.** Transparent agent-identity tokens are in the market and may become the default. Mitigation: the identity layer is in production with a customer; the seller rail needs no new cryptography; the commercial rail is decoupled from every financial-layer milestone; adoption is measured honestly, from independent parties only.
- **Concentration of the identity link at issuers.** The issuer is the only party that can connect an agent to its person. Mitigation: that link never leaves the issuer's vault except under due process or consent, with a receipt on chain; issuers are bonded and council-supervised; a compromised issuer fails closed in under a minute under the freshness rule (reference values).

13

Appendix — where each claim is specified

THIS PAPER	SPECIFICATION OR DOCUMENT
§3 the eight use cases and their status	the eight `Use-Case-*.md` documents of this corpus; `Agent-KYC-Roadmap.md` §1 for the status of each
§4 what is and is not on the chain; freshness; the ten-second proof; grades; recovery	`SPEC-1` Requirements (privacy, custody, status freshness, presentation lifetime, bounded state); §4, §9.1, §10.5, §12.2; `Use-Case-Human-Identity-Verification.md`
§5 agent credential, scope, unlinkability	`SPEC-2` Requirements (human backing, sibling unlinkability, escrow of the link, delegated scope); §2–§4; `Use-Case-Human-Backed-Agent-Credential.md`
§5 kill switch	`SPEC-1` §9.1 (ack rule), §12.1; `Use-Case-Holder-Self-Revocation.md`
§5 x402 sellers and buyer agents	`Use-Case-x402-Seller-Verification.md`; `Use-Case-x402-Buyer-Agent-Identity.md`; bIP-0002
§6.1 consented commercial disclosure	`SPEC-3` Requirements (rail segregation, disclosure class, consent resolution, delegated consent, anti-coercion, receipts, no continuity); §2, §3, §5, §13; `Use-Case-Consented-Commercial-Disclosure.md`
§6.2 dispute path, arbiter market, no funds	`SPEC-2` Requirements (cascade, dispute ciphertext), §12.1, §12.3; `SPEC-3` §4, §4.1; `SPEC-1` §10.5 (dispute retention record); `Use-Case-Dispute-Cascade.md`; ADR-0036, ADR-0038
§7 competitive positioning	`SPEC-3` Background and no-continuity requirement; `Agent-KYC-Roadmap.md` §3
§8 anchor product, first implementer, traction criterion, the served segment and the one rule on every surface	ADR-0011; ADR-0033; `Genesis-Allocation-and-Transparency-Policy.md` §5; `Agent-KYC-Roadmap.md` §2.8
§8 the relying-party route, verifier admission, operated deployments, the enrolled agent, the operator seat	ADR-0030, ADR-0031; ADR-0032 (proposed); `Use-Case-x402-Seller-Verification.md` §7
§9 token, cap, bands, protocol cuts, no protocol fee to verify, operators' own services	`SPEC-1` R15, R26, §6.1, §6.1.1, open parameters; `Genesis-Allocation-and-Transparency-Policy.md` §1–§2; `foundation/Economic-Simulation-SoW.md`
§10 governance, Foundation, related parties, MiCA	`SPEC-1` §7; `Genesis-Allocation-and-Transparency-Policy.md` §3, §5, §5.1; `foundation/Token-Sale-and-MiCA-Title-II-Plan.md`; ADR-0016, ADR-0017, ADR-0034, ADR-0037
§10 audits as milestones	`SPEC-1` R20 and Phase 6; ADR-0020
§11 status and sequence	`Agent-KYC-Roadmap.md` §1–§2
§12 recourse without funds	`SPEC-3` R-A17, R-A27; ADR-0036

Sources. OpenID Foundation, *Identity Management for Agentic AI* (Tobin South et al., October 2025). IBM Think, *The Practitioner's Guide to Non-Human Identities* (Clark and Forward, February 2026), citing Omdia (December 2024) for

the 50:1 ratio. IDC (December 2025) and Gartner for the agent-population projections, labelled as projections. Juniper Research (2025) and published vendor price lists for verification pricing, labelled as estimates. KYA/KYAPay protocol documentation (kyapay.org, Skyfire): the Skyfire developer documentation (docs.skyfire.xyz: KYA Token, Identity Fields, Common Token Claims; read on 30 October 2026) and the KYAPay profile (IETF Internet-Draft draft-skyfire-kyapayprofile-02, June 2026). Concordium, ERC-8004, Visa and Mastercard public announcements. Chainalysis and industry reports for x402 volume, dated. Every figure is refreshed before any decision relies on it.