

GHOST 2026

Privacy-Native Money for Humans, Machines and AI Agents

A shielded digital-cash and autonomous-payment architecture designed as a privacy-first Zcash alternative for the AI era



Conceptual Protocol Whitepaper | Version 0.9 Research Draft | October 2026

Historical image supplied by the requester. Its inclusion is for editorial and historical context and does not imply endorsement by John McAfee, his estate, Zcash, Electric Coin Company, Zcash Foundation, or any current GHOST organization.

STATUS OF THIS DOCUMENT

This is a new 2026 design proposal. It is not an original John McAfee document, not an official network specification, not a claim that the described protocol has been implemented, and not investment advice. It synthesizes the privacy philosophy and selected architectural themes of the historical 2020 and 2024 GHOST whitepapers, then proposes a new shielded and AI-agent payment architecture for independent review and possible future implementation.

Abstract

GHOST 2026 proposes a privacy-native monetary and settlement protocol for a world in which both people and autonomous software agents routinely purchase data, computation, models, APIs, digital goods, machine services and each other's work. The design deliberately moves beyond the historical GHOST privacy stack based primarily on stealth addressing, RingCT, Bulletproofs and Dandelion++ and defines a new zero-knowledge shielded note system intended to provide Zcash-class transaction privacy while retaining GHOST's proof-of-stake lineage and network-level privacy ambitions. The result is not a clone of Zcash. It is a separate design goal: default-shielded digital cash combined with policy-constrained AI-agent wallets, machine-native HTTP payment adapters, private receipts, delegated spending mandates, selective disclosure and a standards-aware path for x402, AP2, MCP and future agent-commerce protocols.

The core privacy objective is simple: routine payments should not reveal the payer, recipient, transferred amount, memo, agent identity or application-level purpose to the public ledger. At the same time, users and institutions should be able to disclose precisely scoped information to an auditor, counterparty or regulator without making their full financial history public. The core autonomy objective is equally simple: an AI agent may spend only within cryptographically defined authority granted by its owner, and every autonomous payment must be bound to a specific resource, price, recipient, time window and policy context so that prompt injection, confused-deputy attacks and replay cannot silently convert a useful agent into an uncontrolled wallet.

This whitepaper therefore defines GHOST 2026 as a layered system: a shielded UTXO-style ledger; a proof-of-stake consensus and finality layer; a private network propagation layer; an agent delegation and payment-intent layer; an HTTP-native G402 adapter inspired by the emerging 402 payment ecosystem; optional shielded assets for stable-value settlement; private receipts and selective-disclosure keys; and a post-quantum migration framework aligned with standardized cryptography rather than unsupported claims of immediate quantum immunity.

THE ONE-SENTENCE POSITIONING

GHOST 2026 is proposed as private digital cash plus private autonomous-agent commerce: a shielded, proof-of-stake Zcash alternative designed from the beginning for AI agents, APIs, robots and machine-to-machine payments.

Design status legend

Label	Meaning
Historical GHOST	A concept or mechanism documented in the 2020 or 2024 GHOST lineage.
External reference	A capability or design pattern observed in Zcash, x402, AP2, MPP, NIST standards or other public systems.
Proposed GHOST 2026	A new architecture in this whitepaper. It should not be presented as deployed until code, tests, audits and a network upgrade actually exist.
Research target	A performance, cryptographic or interoperability objective requiring benchmarking and independent review before adoption.

What GHOST 2026 is intended to compete on

Dimension	Proposed GHOST 2026 objective
Financial privacy	Default shielded transfers hiding sender relationship, receiver relationship and amount, with encrypted memos and selective disclosure.
AI-agent payments	Native delegation, budgets, endpoint-bound payment intents, private receipts, pay-per-call and autonomous sub-agent spending.
Network privacy	Private transaction propagation, Tor/I2P integration targets, Dandelion-style relay behavior and metadata minimization.
Consensus	Proof of Stake with deterministic finality, preserving GHOST lineage while reducing the energy and latency profile of proof-of-work.
Interoperability	Adapters for HTTP 402/x402-style flows, AP2-style mandates, MCP paid tools and A2A service commerce without exposing full transaction graphs.
Auditability	Incoming/outgoing viewing capabilities and purpose-limited disclosure credentials rather than a binary choice between total publicity and total secrecy.
Post-quantum migration	Explicit cryptographic agility and staged adoption of standardized post-quantum primitives where technically appropriate.

Contents

1. Status, scope and terminology
 2. Historical GHOST lineage: 2020 and 2024
 3. Why the AI era changes privacy money
 4. Design goals and non-goals
 5. Threat model
 6. System architecture
 7. Shielded ledger and note model
 8. Zero-knowledge privacy system
 9. Keys, addresses and selective disclosure
 10. Transaction lifecycle
 11. Network-layer privacy
 12. Proof-of-stake consensus and shielded staking
 13. Monetary model and fee design
 14. AI-agent payment architecture
 15. Delegation, mandates and agent wallets
 16. G402 private HTTP payments
 17. MCP, A2A, AP2 and MPP interoperability
 18. Micropayments, channels and batch settlement
 19. Private agent identity and reputation
 20. Shielded assets and stable-value settlement
 21. Governance and protocol upgrades
 22. Post-quantum migration
 23. Performance and scalability
 24. Wallet and developer experience
 25. GHOST 2026 vs Zcash
 26. GHOST 2026 vs historical GHOST
 27. Security analysis and attack scenarios
 28. Implementation roadmap
 29. Open research questions
 30. Conclusion
- Appendix A. Conceptual data structures
- Appendix B. Example agent-payment flows
- Appendix C. Glossary
- Appendix D. References and source notes

1. Status, Scope and Terminology

1.1 Protocol proposal, not a deployed product

GHOST 2026 is deliberately written as a protocol proposal. Cryptographic designs become trustworthy only after specifications, reference implementations, independent audits, adversarial review, test vectors, reproducible builds and sustained operation under real economic attack. Accordingly, this paper separates requirements from implementation choices. Where a mechanism is described as proposed, it is a target architecture. Where a numerical value is described as a target, it is a design objective rather than a performance claim.

A future team adopting this paper should publish a complete cryptographic specification before mainnet deployment. That specification should define the exact field arithmetic, curves or post-quantum primitives, proof transcript, domain separation, note commitment format, nullifier construction, signature scheme, consensus messages, serialization rules, fee calculation and upgrade activation conditions. None of those details should be left to wallet convention.

1.2 Relationship to GHOST and John McAfee

The historical GHOST project used John McAfee's name and privacy philosophy as a central part of its launch-era identity. The 2020 revised paper described a peer-to-peer privacy-oriented electronic cash system with proof of stake, stealth addressing, RingCT, Particl-derived design elements, Dandelion++, cold staking, governance and a broader privacy ecosystem. A later community whitepaper published in 2024 updated that lineage and retained many of the same themes while modernizing its description of staking, confidential transactions, Bulletproofs, veterans and governance. This 2026 proposal uses those papers only as historical and architectural inputs; it does not claim that McAfee authored, approved or anticipated the new AI-agent and zero-knowledge design presented here.

1.3 Relationship to Zcash

Zcash is used throughout this document as an engineering benchmark because it is the best-known deployed cryptocurrency whose modern shielded architecture uses zero-knowledge proofs to hide transaction details. GHOST 2026 is proposed as an alternative, not a fork, clone or affiliated product. The design intentionally differs in consensus, default privacy policy, machine-payment semantics, delegated AI spending and network metadata objectives. Zcash and associated names remain the property of their respective holders.

1.4 Terminology

Term	Meaning in this paper
Shielded transaction	A transaction whose public consensus data does not reveal the ordinary payer relationship, recipient relationship or transferred amount.
Note	A privately owned value object represented by a commitment in the global shielded state.
Nullifier	A unique spend marker derived from a note secret, used to prevent double spending without revealing which note was consumed.
Agent	Software acting with delegated authority to make decisions or purchases. It may be an LLM-based agent, deterministic service, robot or autonomous workflow.
Mandate	A signed delegation that defines what an agent may buy, from whom, for how much and for how long.
G402	The proposed GHOST 2026 adapter for HTTP 402-style machine payments.
Viewing capability	A key or credential that reveals a scoped subset of otherwise shielded payment information.
Full privacy	A design objective of strong transaction confidentiality and unlinkability; not a guarantee against compromised endpoints, voluntary disclosure, timing correlation or implementation bugs.

2. Historical GHOST Lineage: 2020 and 2024

2.1 What is retained from the historical vision

The most important continuity is philosophical rather than code-level: ordinary financial activity should not become a permanent public dossier. The 2020 revised GHOST design emphasized proof of stake, transaction confidentiality, recipient privacy, network propagation privacy, cold staking and a wider privacy ecosystem. The later 2024 community document continued the proof-of-stake and confidential-transaction direction and documented a mature vocabulary around staking, RingCT, commitments, Bulletproofs, Dandelion++, governance and veteran participation. GHOST 2026 retains the emphasis on privacy, proof of stake, network metadata protection, delegated participation and community governance.

2.2 What is replaced

The central technical change is the privacy engine. Ring signatures, RingCT and Bulletproof-style range proofs can provide meaningful confidentiality, but the user asked for a direct privacy alternative to Zcash. GHOST 2026 therefore proposes a shielded note system in which a succinct zero-knowledge proof establishes ownership, conservation of value and correct state transition without publicly exposing the spent notes, recipient or amount. RingCT is retained as historical context, not as the primary 2026 transfer primitive.

2.3 What is extended for the AI economy

The historical papers were written before autonomous software routinely negotiated with APIs, called paid tools, delegated sub-tasks and spent money on behalf of users. In 2025 and 2026 the payment industry began standardizing machine-native payment flows around HTTP 402, agent mandates, verifiable user intent and tool-level commerce. GHOST 2026 treats those capabilities as first-class protocol requirements. The key addition is not simply that an agent can hold a wallet. The protocol must prove that the agent acted inside a tightly constrained authority envelope and that the payment was bound to the intended resource.

Historical component	2026 treatment	Reason
Proof of Stake	Retained and redesigned with deterministic finality	Continuity with GHOST and lower-latency settlement.
Cold staking	Retained conceptually as delegated staking keys	Separates spending authority from validator participation.
Stealth addresses	Absorbed into shielded address/note design	A shielded note system makes ordinary receiver addresses invisible on-chain.
Ring signatures / RingCT	Replaced as primary privacy primitive	GHOST 2026 targets ZK shielded privacy closer in spirit to Zcash.
Bulletproofs	Optional component/research tool rather than core transfer proof	The base shielded system uses a general zero-knowledge validity proof.
Dandelion++	Retained and expanded	Ledger privacy is weakened if IP and timing metadata remain easy to observe.
Ghost Veterans	Reinterpreted as optional validator/governance participation tier	Preserves community lineage without requiring a conventional masternode model.
MetaGhost / broader app vision	Extended into agent wallets, G402 and private service payments	The wallet becomes an autonomous economic policy engine.

3. Why the AI Era Changes Privacy Money

3.1 Agents are becoming economic actors

A human payment is usually an occasional explicit act. Agentic payment behavior is different. A research agent may buy dozens of data points, a coding agent may pay for tool calls, a shopping agent may negotiate across merchants, a robot may purchase charging or mapping data, and an enterprise workflow may spawn sub-agents that hire specialized services. The number of transactions can grow dramatically while the amount of each transaction falls. The payment system must therefore handle high-frequency low-value settlement, programmable authority and machine-readable receipts.

Emerging protocols confirm this direction. x402 uses HTTP 402 as a machine-payment trigger, while Google's AP2 adds signed mandates and evidence of user authorization for agent-led purchases. Cloudflare documents both x402 and MPP as mechanisms for agentic payments, including paid MCP tools. GHOST 2026 does not attempt to replace these application-level standards. It attempts to become a privacy-preserving settlement rail beneath them.

3.2 Public ledgers are especially invasive for agents

A transparent agent wallet can reveal far more than a normal consumer wallet. The graph may identify the model providers used by a company, the data sources consulted before a trade, the suppliers a procurement agent is evaluating, the medical APIs called by a health workflow, or the exact tools a research organization uses. Even if names never appear on-chain, transaction timing and counterparties can expose strategy. A privacy-preserving agent economy therefore requires hiding both money flows and service-selection metadata.

3.3 Autonomous spending increases authorization risk

Giving an AI agent a normal private key is equivalent to giving software an unrestricted bearer instrument. That model is unsafe. Prompt injection, compromised tools, malicious websites, hallucinated addresses and confusing merchant flows can all redirect funds. GHOST 2026 separates an owner root key from constrained agent capability keys. The agent can prove that a payment is authorized under a signed mandate without obtaining unrestricted control of the owner's entire balance.

3.4 Privacy and accountability must coexist

Institutional adoption requires a third property beyond privacy and autonomy: explainability. A business may need to prove to an auditor that an agent spent less than its daily allowance, paid an approved provider, or purchased a permitted class of service. Public disclosure of the complete transaction graph is not necessary for those proofs. GHOST 2026 therefore treats selective disclosure and zero-knowledge policy proofs as core capabilities rather than afterthoughts.

4. Design Goals and Non-Goals

4.1 Primary goals

- Default shielded value transfer for ordinary wallet-to-wallet and agent-to-service payments.
- No public exposure of ordinary sender-recipient links, transferred amount or encrypted memo contents.
- Low-friction selective disclosure using separate incoming, outgoing and audit-oriented viewing capabilities.
- Proof-of-stake consensus with rapid deterministic finality suitable for machine commerce.
- Private network propagation that makes ledger privacy harder to defeat through simple IP or timing observation.
- Native delegated authority for AI agents with budgets, merchant constraints, service constraints, expiry, revocation and sub-delegation rules.
- Resource-bound payment proofs so a valid payment credential cannot be replayed for a different endpoint, item or service.
- Compatibility adapters for HTTP 402/x402-style systems, AP2-like authorization models, MCP tool payments and A2A commerce.
- Cryptographic agility, including a staged post-quantum migration path aligned with standardized primitives.
- No trusted operator able to decrypt the global payment graph.

4.2 Non-goals

- Promising perfect anonymity under all conditions. Endpoint compromise, malware, browser fingerprinting, merchant logs, timing analysis and user disclosure can defeat privacy outside the ledger.

- Embedding unrestricted general-purpose smart contracts in the base layer at launch. The initial proposal favors narrowly defined private payment covenants because they are easier to audit.
- Claiming instant post-quantum security. Zero-knowledge systems and wallet signatures require careful redesign before a complete PQ transition.
- Providing a mechanism to evade lawful obligations. Selective-disclosure tools are intended to let users satisfy legitimate reporting needs without publishing unrelated financial history.
- Guaranteeing exchange listing, market value, investment return or regulatory treatment.
- Pretending that a whitepaper is equivalent to tested software.

5. Threat Model

5.1 Adversaries considered

Adversary	Capability	Desired resistance
Public blockchain observer	Reads every block, mempool artifact and consensus object	Should not derive ordinary payer/recipient/amount relationships from shielded transfers.
Network observer	Monitors many peers, timing and IP metadata	Should face additional uncertainty from private routing, delayed stem propagation and anonymity-network support.
Malicious merchant	Controls a paid API or storefront	Should not be able to overcharge, replay, substitute resources or turn a narrow mandate into arbitrary withdrawal authority.
Compromised AI tool	Injects instructions into an autonomous agent	Should be blocked by an independent wallet policy engine that evaluates signed mandates rather than model text alone.
Compromised agent process	Steals agent capability key	Should be limited by budget, recipient, time, service and rate constraints, and revocation should limit future loss.
Validator cartel	Censors or reorders transactions	Should be limited by distributed consensus, slashing and censorship-detection mechanisms; privacy should reduce targeted economic discrimination.
Malicious wallet software	Leaks metadata or keys	Cannot be fully solved by consensus; reproducible builds, hardware isolation and external audits are required.
Future quantum adversary	Breaks vulnerable public-key assumptions	Protocol should support planned key migration and hybrid deployment of standardized PQ primitives.

5.2 Security invariants

1. Conservation: a valid transaction cannot create spendable native value except through consensus-authorized issuance.
2. No double spend: every consumed note maps to a unique nullifier and duplicate nullifiers are rejected.
3. Authorization: spending requires knowledge of the appropriate spending secret or a valid constrained delegation.
4. Policy binding: an agent transaction must satisfy the active mandate constraints proven or verified by the wallet/protocol.
5. Resource binding: an autonomous payment credential is cryptographically committed to the intended resource, endpoint, quote and nonce.
6. Receipt binding: a private receipt cannot be transplanted to prove purchase of a different resource.
7. Shielding: public consensus data should not directly reveal note ownership, recipient identity or transferred amount.
8. Disclosure isolation: a scoped viewing credential should reveal only the intended subset of transaction information.
9. Upgrade safety: protocol upgrades must not silently weaken the privacy set of existing notes.

10. Revocation: revoking an agent capability must prevent future use after a defined propagation/finality boundary.

6. System Architecture

6.1 Layered architecture



6.2 Core components

Component	Responsibility
GHOST Shielded Ledger (GSL)	Maintains note commitment state, nullifier set, shielded assets and proof verification.
GHOST Proof System (GPS)	Proves ownership, value conservation, output correctness and optional policy constraints without exposing private inputs.
GHOST Stake Finality (GSF)	Proof-of-stake validator selection, block proposal, attestation, deterministic finality and slashing.
GHOST Privacy Transport (GPT)	Transaction propagation, anonymity-network integration, connection rotation and anti-correlation controls.
GHOST Agent Mandates (GAM)	Defines delegated authority, budgets, service scopes, expiry, revocation and sub-delegation.
G402	Maps HTTP 402 payment challenges into resource-bound shielded payments and private receipts.
Ghost View & Audit Keys (GVAK)	Supports incoming/outgoing viewing and purpose-limited disclosure.
Ghost Shielded Assets (GSA)	Optional multi-asset notes for private stable-value and token settlement under protocol governance.

6.3 Separation of policy from the language model

The model that decides what to buy should not be the component that unconditionally signs the payment. GHOST 2026 uses a two-process mental model. The agent proposes a purchase. A wallet policy engine validates the signed owner mandate, merchant identity or credential, resource digest, price, frequency, budget, expiry and recent spending. Only then may the wallet create a spend proof. This separation is analogous to privilege separation in operating systems: a potentially manipulable reasoning process does not receive unrestricted financial authority.

7. Shielded Ledger and Note Model

7.1 UTXO-style private notes

The proposed ledger represents private value as notes rather than public account balances. Each note commits to a value, asset identifier, recipient key material and fresh randomness. The commitment is inserted into an append-only commitment tree. A spender proves in zero knowledge that the consumed note is a member of an accepted historical root, that the spender is authorized, and that the transaction balances. The note itself does not appear in plaintext on-chain.

Conceptual note commitment:

```
cm = Commit(asset_id || value || recipient_data || rho || randomness)
```

Public chain stores: cm, proof-related data, encrypted note ciphertext and later a nullifier when spent.

Private wallet stores: decrypted note plaintext and spending/viewing key material.

7.2 Nullifiers and double-spend prevention

A private payment cannot reveal which note is consumed, but consensus must still reject a note spent twice. Each note therefore derives a nullifier from note-specific secret material and the owner's spend authorization key. The zero-knowledge proof establishes that the published nullifier corresponds to a valid committed note without disclosing that note. The nullifier becomes public only when the note is spent and cannot be used to identify the note unless another secret is leaked.

7.3 Value conservation

For each asset inside the shielded pool, the proof must establish that the sum of consumed value plus authorized issuance equals the sum of created value plus fees or authorized burns. This statement is proven without publishing individual amounts. Native GHOST issuance is limited to consensus-defined validator rewards or other protocol-authorized mechanisms.

7.4 Encrypted note delivery

A recipient needs enough encrypted metadata to discover and decrypt incoming notes. The transaction therefore carries encrypted note ciphertexts that can be trial-decrypted by wallets using view-key-derived material. To reduce scanning cost, the final design may use compact tags, diversified addresses or other receiver hints that do not create stable public identifiers. This choice must be evaluated for privacy leakage before standardization.

7.5 Default shielding policy

The proposed user experience is shielded by default. Unlike a system in which users must understand multiple address types to obtain strong privacy, the normal GHOST 2026 wallet should create shielded notes for ordinary transfers and agent payments automatically. Transparent transfer support, if retained for migration or exchange compatibility, should be visibly marked as a legacy or disclosure mode and should not be the default receiving path.

8. Zero-Knowledge Privacy System

8.1 Requirements for the proving system

- No per-application trusted setup if feasible; the preferred design is a modern setup-free or universal-setup system whose security assumptions are clearly documented.
- Succinct proofs with validator verification significantly cheaper than re-executing the private transaction logic.
- Practical proving on modern desktop and mobile hardware, with optional hardware acceleration.
- Batch verification or recursive aggregation to support high-frequency agent payments.
- Circuit upgradability under explicit network-upgrade rules, not hidden server-side changes.
- Well-defined domain separation and transcript construction to prevent cross-protocol proof reuse.
- Open-source reference implementations in at least two independently maintained codebases before high-value deployment.

8.2 Zcash as a benchmark, not a dependency

Zcash demonstrates that a production network can use succinct zero-knowledge proofs to validate shielded payments. Its modern Orchard design uses Halo 2 and a note/nullifier architecture that avoids directly publishing ordinary transaction relationships. GHOST 2026 adopts the requirement - prove a valid private state transition without revealing the transaction graph - but does not require copying Orchard or using Zcash code. A future GHOST cryptography working group should compare Halo 2, newer Plonkish systems, proof recursion frameworks and post-quantum-oriented proof systems before selecting the final engine.

8.3 Core shielded proof statement

A GHOST 2026 spend proof should establish, at minimum:

1. Every input note commitment exists under an accepted commitment-tree root.
2. The prover has valid spend authority for every input.
3. Each public nullifier is correctly derived from a consumed private note.
4. No hidden value is created: inputs + authorized issuance = outputs + fee + authorized burn.
5. Every output note is correctly formed and encrypted to the intended receiver.
6. Any active covenant or agent mandate constraint is satisfied.
7. The proof is bound to this transaction digest, network identifier and protocol version.

8.4 Private memos

Memos are valuable in machine commerce because an agent may need an invoice identifier, model-call reference, service-level agreement hash or private receipt pointer. GHOST 2026 treats memos as encrypted payloads rather than public text. Wallets should distinguish human-readable memos from protocol-bound receipt data and avoid putting personally identifying information into permanent ciphertext when shorter-lived off-chain references are sufficient.

8.5 Anonymity set discipline

Strong cryptography does not automatically produce strong anonymity if wallet behavior fragments users into small identifiable subgroups. The protocol should minimize unnecessary transaction modes, discourage distinctive change patterns, standardize common fee tiers and support transaction padding where the privacy benefit justifies the bandwidth cost. Default shielding is therefore not merely a UX preference; it is a mechanism for enlarging the effective anonymity set.

9. Keys, Addresses and Selective Disclosure

9.1 Key hierarchy

Key / capability	Purpose	Should it spend funds?
Master spending key	Derives account spending authority and recovery structure	Yes; highest sensitivity.
Full viewing key	Discovers incoming and outgoing activity for an account	No.
Incoming viewing key	Discovers payments received by a scoped account	No.
Outgoing disclosure key	Reconstructs or proves selected outgoing payment data	No.
Agent capability key	Authorizes only transactions inside a signed mandate	Only within policy.
Stake authorization key	Participates in staking without exposing ordinary spend key	No ordinary wallet spending.
Audit disclosure credential	Reveals a selected account, time period, asset, counterparty class or proof of policy compliance	No.

9.2 Unified private addresses

The ordinary user should not need to understand whether a payment uses a stealth address, shielded receiver or agent receiver. A GHOST unified address may encode multiple receiver capabilities under a single human-facing string while wallets prefer the strongest supported private receiver. The address itself should not become a permanent public identifier on-chain; transaction outputs should use fresh note destinations derived from it.

9.3 Selective disclosure modes

- Incoming-only disclosure for donations, revenue verification or proof of receipt.
- Outgoing-only disclosure for expense review without exposing all incoming funds.
- Time-bounded audit disclosure for a specified period.
- Counterparty-bounded disclosure for transactions involving a designated merchant or institutional wallet.
- Amount-range proofs that demonstrate an expense was below or above a threshold without revealing the exact amount.
- Balance or solvency proofs that demonstrate control of sufficient private assets without publishing all notes.
- Agent-policy proofs showing that every payment in a review set complied with a signed budget mandate.

9.4 Disclosure should be deliberate

Viewing capabilities must be clearly separated in wallet UX. Sharing a full viewing key can reveal far more than a one-time transaction disclosure. The wallet should generate human-readable summaries of what a disclosure token will expose before export. Enterprise software should log disclosure grants locally and support revocation of off-chain access, while acknowledging that a recipient who has already learned data cannot be forced to forget it.

10. Transaction Lifecycle

10.1 Human-to-human shielded payment

1. The sender wallet resolves the recipient's unified private address into a shielded receiver.
2. The wallet selects private input notes and obtains a recent accepted commitment-tree root.
3. It creates one or more private output notes, including change, and encrypts note plaintexts to recipients.
4. It derives nullifiers for the consumed inputs and constructs the zero-knowledge witness.
5. It creates a proof of membership, authorization, nullifier correctness and value conservation.
6. It signs the transaction authorization and sends the transaction through the privacy transport layer.
7. Validators verify the proof, signatures, nullifier uniqueness, fee rule and consensus validity.
8. After deterministic finality, recipient wallets discover and decrypt the new notes.

10.2 Agent-to-service payment

1. An agent requests a resource or tool call.
2. The provider returns a machine-readable quote or HTTP 402 challenge with price, resource digest, expiry, nonce, accepted network and recipient descriptor.
3. The agent proposes the purchase to its wallet policy engine.
4. The policy engine verifies the owner mandate, merchant constraints, budget, quote integrity and anti-replay state.
5. The wallet creates a shielded payment whose private receipt commitment includes the resource digest and quote nonce.
6. The service verifies a payment credential or facilitator-confirmed private receipt without learning unrelated wallet history.
7. The provider returns the requested resource and a fulfillment receipt.
8. The agent records an encrypted local audit event linking purchase intent, payment receipt and service result.

10.3 Offline and delayed settlement

Not every machine can wait for finality. GHOST 2026 therefore distinguishes base-layer settlement from payment authorization. Small trusted interactions may use signed, resource-bound vouchers or payment-channel updates that are later settled in batches. The merchant chooses a risk policy: accept immediately below a threshold, wait for one finalized block, or require channel collateral. The protocol should never pretend that an unsettled voucher is the same as finalized on-chain value.

11. Network-Layer Privacy

11.1 Why blockchain privacy is not enough

A perfectly shielded transaction can still leak useful metadata if an observer sees the originating IP address at the moment the transaction first appears. Historical GHOST documentation recognized this problem through Dandelion++. GHOST 2026 elevates network privacy to a required subsystem rather than an optional wallet feature.

11.2 Proposed privacy transport

- Stem-and-fluff propagation inspired by Dandelion-style routing for ordinary transactions.
- Native optional routing over Tor and I2P with wallet defaults that favor privacy when latency permits.
- Encrypted peer transport with authenticated protocol negotiation.
- Connection rotation and peer-selection rules designed to reduce long-term topology fingerprinting.
- Randomized propagation delay within bounded windows to reduce trivial first-seen correlation.
- Batching and cover-traffic research for high-risk deployments, while measuring the bandwidth and denial-of-service cost.
- Separate relay paths for block propagation and transaction origin privacy so performance optimization does not silently disable privacy.

11.3 Metadata limitations

PRIVACY LIMIT

No network-routing mechanism can guarantee anonymity against a global passive adversary observing both ends of communication. Wallets should state this clearly. GHOST 2026 aims to raise the cost of correlation and remove avoidable leaks, not to market mathematically absolute network anonymity.

12. Proof-of-Stake Consensus and Shielded Staking

12.1 Consensus philosophy

Proof of stake is one of the clearest continuities with historical GHOST. For 2026, the proposed consensus objective is rapid deterministic finality rather than long probabilistic confirmation. The exact BFT protocol is intentionally left open pending benchmark and formal analysis, but the validator system should use epoch-based stake selection, explicit slashing for provable equivocation, bounded unbonding and protocol-governed parameter changes.

12.2 Proposed roles

Role	Function
Validator	Proposes or attests blocks and participates in finality.
Delegator	Assigns staking weight without transferring ordinary spending authority.
Ghost Veteran (optional lineage role)	Long-lived validator/delegator meeting higher reliability and governance requirements; no special surveillance privilege.
Light client	Verifies finalized headers and shielded proofs relevant to its own wallet without running a full archival node.

12.3 Shielded stake considerations

A privacy coin faces a tension: consensus needs to know that voting power exists, while users may not want to publish their full holdings. One research direction is a staking certificate that proves a locked or bonded position exists without reusing the ordinary payment address. At minimum, staking and spending keys should be separate, and the protocol should avoid forcing users to reveal their complete transaction history simply to participate in consensus.

12.4 Target timing

Historical GHOST used much slower block intervals than an agentic payment network ideally needs. GHOST 2026 therefore sets a research target of 4-8 second block slots and deterministic economic finality in roughly 12-24 seconds under normal network conditions. Small machine payments can achieve lower perceived latency through channels or accepted pre-finality vouchers. These numbers are targets, not claims of implemented performance.

13. Monetary Model and Fee Design

13.1 Supply continuity

The historical GHOST lineage used a 55 million GHOST supply target. This paper recommends retaining that number if the 2026 protocol is adopted as an upgrade or migration of the existing network, because arbitrary monetary reset would undermine continuity. The exact circulating supply, remaining emission and validator reward schedule must be derived from verifiable chain state at the activation point. This whitepaper does not authorize a new premine, token sale or discretionary reissuance.

13.2 Migration options

Option	Description	Trade-off
In-place protocol upgrade	Existing chain activates new shielded and agent-payment features through network upgrade	Best supply continuity; hardest engineering migration.
Parallel shielded pool	Existing consensus adds a new GHOST 2026 shielded pool and gradually deprecates older transfer modes	Practical staged migration; temporarily fragments privacy sets.
New chain with cryptographic claims	A clean-slate chain lets historical holders claim balances using proof of ownership	Simpler architecture but governance, replay and legitimacy risks are substantial.

13.3 Fee model

AI commerce requires tiny payments, so a fee market designed only for large retail transfers is insufficient. The fee model should price scarce resources - proof verification, encrypted payload bytes, state growth and block space - while avoiding unique fee fingerprints that weaken privacy. One approach is standardized fee buckets. Wallets round fees into a small set of common classes and the proof demonstrates sufficient fee funding without exposing the user's payment amount.

13.4 Microunits

Wallets should support human-readable decimal GHOST while APIs operate on integer base units. The exact decimal precision should be standardized once and remain stable. For agent commerce, SDKs should expose safe fixed-point arithmetic and reject floating-point payment values.

14. AI-Agent Payment Architecture

14.1 Design principle: agents receive authority, not unrestricted wallets

The owner does not simply hand an agent a seed phrase. Instead, the owner signs a mandate defining the economic authority available to that agent. The agent receives a capability key that can authorize only transactions consistent with the mandate. The wallet policy engine is responsible for enforcement even if the reasoning model asks for something broader.

14.2 GHOST Agent Mandate

Conceptual mandate fields:

mandate_id	random unique identifier
owner_key	root authorization identity

agent_key	delegated capability key
asset_scope	GHOST and/or allowed shielded assets
max_per_payment	upper bound per transaction
max_per_period	budget per hour/day/week
merchant_scope	allowlist, credential class or wildcard rules
service_scope	API/data/compute/tool/product categories
valid_from / valid_until	validity window
max_subdelegation_depth	whether the agent may create constrained sub-agents
network_scope	permitted chains or settlement domains
require_receipt	whether fulfillment proof is mandatory
revocation_nonce	state used for revocation
owner_signature	signature over canonical mandate digest

14.3 Policy evaluation

Policy is evaluated against structured data, not natural-language intent. The agent may say, “buy the cheapest market-data API,” but the wallet should evaluate the actual merchant credential, endpoint digest, quoted amount, currency, expiration and cumulative budget. If any field is outside the mandate, the wallet refuses or requests human approval.

14.4 Sub-agent delegation

An orchestrator may hire a specialist agent. Sub-delegation is allowed only if the parent mandate explicitly permits it. A child mandate must be strictly narrower than its parent: lower or equal budget, shorter expiry, equal or narrower merchant scope and lower delegation depth. A zero-knowledge policy proof may eventually allow a service provider to verify that a child payment is properly delegated without learning the entire delegation chain.

14.5 Agent treasury separation

Enterprises should be able to fund multiple agent spending pools without exposing the organization's total holdings. A treasury wallet can create shielded policy accounts with independent budgets and viewing scopes. Finance staff may hold a review key for one department while the corporate root wallet remains private.

15. Delegation, Mandates and Agent Wallets

15.1 Three-key operating model

Key	Where it lives	Authority
Owner root key	Hardware wallet, secure enclave or multisig policy	Creates/revokes mandates; may recover funds.
Agent capability key	Agent runtime or isolated signing service	Initiates mandate-compliant payments only.
Policy attestation key	Independent wallet/policy engine	Attests evaluated quote/resource context; can require additional human approval.

15.2 Defense against prompt injection

Prompt injection is not solved by asking the same model whether a payment is safe. The payment boundary must be structurally separate. The agent produces a proposed purchase object; a deterministic policy layer checks the canonical fields; optional external risk modules evaluate merchant credentials and recent behavior; only then can the capability key authorize a transaction. High-risk actions can require a hardware-wallet confirmation or a second agent operating under a different model and policy.

15.3 Revocation

Mandates include a revocation identifier. A revocation transaction or finalized revocation registry update invalidates future use. Wallets should also support immediate local revocation before chain finality, preventing the user's own signer from producing new payments while the network state catches up. Short-lived capability keys reduce the value of stolen credentials.

15.4 Human-not-present transactions

AP2 and similar industry work distinguish transactions where a human is present from autonomous transactions made under prior authority. GHOST 2026 adopts the same conceptual distinction. A “human present” payment may use a one-time approval. A “human not present” agent payment requires an active mandate with explicit limits. The receipt should record which authorization mode applied, encrypted to the parties and any designated auditor.

16. G402: Private HTTP 402 Payments

16.1 Motivation

HTTP 402 is emerging as a convenient interface for machine-native payment. The server can respond to an unauthenticated request with a machine-readable challenge; the client pays; then it retries with proof of payment. Existing implementations commonly settle transparent stablecoin transfers. G402 keeps the request/response ergonomics while using shielded GHOST settlement and resource-bound private receipts.

16.2 Challenge format

Illustrative G402 challenge:

```
HTTP/1.1 402 Payment Required
GHOST-Payment-Version: 1
GHOST-Network: mainnet
GHOST-Asset: GHOST
GHOST-Price-Commitment: <commitment or encrypted quote>
GHOST-Pay-To: <shielded receiver descriptor>
GHOST-Resource-Digest: <hash of endpoint + parameters + quote terms>
GHOST-Nonce: <server nonce>
GHOST-Expires: <timestamp>
GHOST-Receipt-Key: <ephemeral receipt encryption key>
```

16.3 Payment credential

The retry carries a compact credential proving that the client created or committed to a valid payment satisfying the challenge. The credential is bound to the server nonce, resource digest, price terms, merchant receiver and expiry. A facilitator may confirm settlement without learning the client's full wallet history. The exact protocol must avoid the substitution and replay weaknesses that researchers have identified in early machine-payment designs.

16.4 Privacy choices

The merchant necessarily learns enough to know that its own invoice was paid. It should not learn the payer's unrelated transactions, total wallet balance or stable public address. Where legal or business policy requires customer identification, that identity should travel through an explicit credential channel rather than be inferred from a permanently transparent payment graph.

17. MCP, A2A, AP2 and MPP Interoperability

17.1 Interoperability philosophy

Agent commerce is developing faster than any single blockchain standard. GHOST 2026 should therefore expose adapters rather than force developers into a proprietary agent framework. The settlement rail remains GHOST; application protocols can change independently.

External protocol	GHOST 2026 role
x402-style HTTP payments	G402 maps a 402 challenge to a shielded payment and private receipt.
MPP	A machine-payment credential can carry or reference a GHOST shielded settlement proof.

AP2	Intent and payment mandates can be mapped to GHOST Agent Mandates and receipt evidence.
MCP	Paid tools can advertise G402 payment requirements per tool call.
A2A	Agents can negotiate service terms, then settle privately through GHOST.

17.2 Mandate mapping

An AP2-style intent mandate describes user authorization at the commerce layer. A GHOST Agent Mandate governs blockchain spending authority. An adapter should bind the two digests so that a payment cannot be detached from the authorized purchase. The chain does not need to understand every commerce field; the wallet and receipt circuit need only commit to the canonical purchase digest and prove that the spending policy allowed it.

17.3 Paid MCP tools

A private MCP server may declare that a tool call costs a certain amount. The agent receives the tool schema and payment challenge, obtains wallet authorization, attaches a G402 credential and receives the result. For repetitive calls, the agent and server should prefer a private channel or batch voucher rather than generate a full base-layer transaction for every inference or data lookup.

18. Micropayments, Channels and Batch Settlement

18.1 Why base-layer payment per token is inefficient

AI services increasingly charge at extremely fine granularity. Settling every model token, database row or sensor reading as a separate shielded transaction would waste block space and create unnecessary proof work. GHOST 2026 therefore defines multiple settlement modes while preserving a single economic unit.

Mode	Best for	Privacy/settlement property
Direct shielded payment	Retail transfers and medium-value service calls	Immediate on-chain finality after consensus.
Pre-funded private channel	Repeated calls between the same parties	Only channel open/close or periodic checkpoints need chain settlement.
Batch voucher settlement	Thousands of small agent purchases	Provider aggregates signed vouchers and settles a net amount.
Probabilistic micropayment (research)	Extremely low-value streams	Reduces settlement frequency but requires careful fairness analysis.

18.2 Channel privacy

A private channel should avoid exposing every intermediate balance update. The two parties exchange signed state commitments off-chain. The final settlement proves the valid latest state or uses a dispute window. For shielded systems, channel design is technically difficult because a public dispute can leak relationship metadata. The first implementation should favor simple bilateral channels with rigorous auditing over a feature-rich generalized state-channel VM.

18.3 Batch settlement safety

Batch settlement must bind every voucher to merchant, resource and nonce; otherwise one proof can be replayed to claim unrelated service. The batch root should commit to ordered voucher digests, and each voucher should include an expiry and unique identifier. Merchant software must enforce atomic accounting so concurrent requests cannot consume the same authorization more than once.

19. Private Agent Identity and Reputation

19.1 Identity without a permanent payment address

Agents need reputation, but reputation should not require reusing a transparent wallet address. A GHOST agent can present a credential proving membership in a reputation domain, successful completion count, stake bond or service category without revealing its complete payment history. The credential may be issued by a marketplace, enterprise or decentralized reputation system.

19.2 Pseudonymous continuity

A user may choose persistent pseudonymous identity for a merchant relationship while using unlinkable shielded payment notes underneath. This separates business reputation from transaction graph exposure. A merchant can know that the same certified agent returned without learning which other merchants that agent paid.

19.3 Sybil resistance

No single identity primitive eliminates Sybil attacks. GHOST 2026 supports several optional economic proofs: bonded stake, credentialed membership, rate-limited anonymous credentials and reputation attestations. The system should avoid requiring global real-name identity merely to make a private payment.

20. Shielded Assets and Stable-Value Settlement

20.1 Why agents may need stable units

The native GHOST asset provides fees, security and settlement, but many machine services prefer a stable unit of account. GHOST 2026 therefore proposes a generic shielded asset note format. An approved asset identifier can represent a stable-value token or other instrument inside the same privacy machinery. Asset issuance itself must be separately governed; the base protocol should not pretend that a stable asset is solvent merely because its transfers are private.

20.2 Multi-asset privacy considerations

Asset identifiers can fragment anonymity. If every obscure token has only a few users, hiding sender and receiver may still leave a small behavioral set. Wallets should communicate this limitation and may use batching or common-denomination techniques where appropriate. Cross-asset swaps add another linkability surface and should use private swap protocols rather than transparent bridge addresses when possible.

20.3 Native GHOST role

GHOST remains the protocol's native security and fee asset. Shielded stable assets are optional payment instruments, not replacements for GHOST's consensus role. Validator rewards, slashing collateral and protocol-level fee accounting should be denominated primarily in GHOST unless governance explicitly adopts a different model.

21. Governance and Protocol Upgrades

21.1 Governance objectives

- No single company should be able to silently alter the monetary rules or decrypt the shielded pool.
- Cryptographic upgrades must be public, versioned, testable and activated only after defined review and signaling periods.
- Privacy regressions must be treated as consensus-level risks, not ordinary UX changes.
- Emergency responses should prefer client updates and coordinated network activation over permanent administrator keys.
- Funding mechanisms, if any, should be transparent in consensus rules even when individual user transfers are private.

21.2 Ghost Veterans as governance participants

If the Veteran concept is retained, it should be a transparent participation role rather than a privileged surveillance or custody tier. Eligibility might depend on bonded stake, uptime and long-term participation. Governance weight should be capped or balanced so that a small number of wealthy Veterans cannot unilaterally control upgrades.

21.3 Upgrade process

1. Publish a draft improvement proposal with security rationale, privacy impact and migration plan.
2. Provide executable test vectors and reference implementation branches.
3. Commission independent cryptographic and consensus review for major changes.
4. Run public testnet and adversarial testing period.
5. Collect ecosystem readiness signals from validators, wallets and infrastructure providers.
6. Activate at a predetermined finalized height or epoch with explicit rollback contingency limited to catastrophic pre-activation bugs.

22. Post-Quantum Migration

22.1 Update to the historical quantum-resistance discussion

Earlier privacy-coin documents often used the phrase “quantum resistant” too loosely. By 2026, a responsible design can be more precise. NIST has standardized ML-KEM for key establishment and ML-DSA plus SLH-DSA for signatures. These standards provide useful building blocks, but replacing a cryptocurrency's full cryptographic stack is more complex than swapping one signature algorithm. Shielded proof systems, address formats, wallet recovery and consensus signatures all need separate analysis.

22.2 Proposed migration phases

Phase	Goal
PQ-0: Cryptographic agility	Version key formats, domain separation and transaction encodings so new primitives can be added without redesigning the whole ledger.
PQ-1: Hybrid transport and messaging	Use ML-KEM in optional hybrid encrypted channels and wallet-to-wallet metadata exchange where practical.
PQ-2: Hybrid account/validator signatures	Permit classical + ML-DSA or other standardized signature combinations for new long-lived keys after performance review.
PQ-3: Shielded-proof migration research	Evaluate proof systems whose core assumptions and performance meet post-quantum objectives.
PQ-4: Mandatory migration window	If needed, require funds to move to new note/key formats before a defined deprecation height.

22.3 Honest terminology

DO NOT MARKET PREMATURELY

Until the shielded proof system, transaction authorization and consensus signatures all use appropriately reviewed post-quantum assumptions, the network should say “post-quantum migration ready” or “cryptographically agile,” not “fully quantum proof.”

23. Performance and Scalability

23.1 Performance objectives

Metric	Research target	Notes
--------	-----------------	-------

Wallet proof generation	Sub-second desktop for common transfer; under several seconds on modern mobile	Depends on final proof system and hardware acceleration.
Validator proof verification	High-throughput batch verification	Must remain cheaper than re-executing private state.
Block slot	4-8 seconds	Subject to network simulations and BFT safety.
Deterministic finality	12-24 seconds under normal conditions	Channels/vouchers handle lower-latency agent UX.
Agent micropayments	Thousands of off-chain voucher events per settled batch	Avoids one base-layer tx per API call.
Wallet sync	Compact encrypted scanning with incremental state and checkpoints	Critical for mobile adoption.

23.2 Proof aggregation

Recursive or batch proof aggregation can reduce the number of expensive objects validators verify. An aggregator should not become a trusted privacy intermediary; anyone should be able to aggregate valid proofs. The base protocol should verify the aggregated result and preserve the ability to fall back to individual verification for debugging or dispute analysis.

23.3 State growth

Shielded note commitment trees and nullifier sets grow permanently. Nodes need pruning, compact checkpoints and efficient authenticated data structures. Light wallets should not download every encrypted note body indefinitely. Protocol designers must measure whether privacy-preserving scan hints leak linkable receiver information before adopting them.

24. Wallet and Developer Experience

24.1 Human wallet

- One primary private address with automatic strongest-receiver selection.
- Clear labels when a legacy or transparent path would reduce privacy.
- Hardware-wallet support for root keys and mandate approval.
- One-click generation of scoped disclosure reports.
- Built-in privacy transport status showing whether Tor/I2P/private relay is active.
- Privacy-aware address book that does not encourage permanent on-chain identifiers.
- Safe recovery that separates spending keys, viewing keys and agent capability keys.

24.2 Agent wallet SDK

- `createMandate()` / `revokeMandate()`
- `proposePurchase(quote, resourceDigest)`
- `evaluatePolicy(mandate, quote, recentSpend)`
- `payShielded(receiver, amount, receiptBinding)`
- `openMicropaymentChannel(provider, limit)`
- `exportPrivateReceipt(scope)`
- `generateDisclosureProof(statement)`
- `getRemainingBudget(mandateId)`

24.3 Safety defaults

Developer convenience must not turn into silent financial authority. SDKs should default to low test limits, explicit mainnet opt-in, maximum payment caps, deterministic canonical serialization and simulated policy evaluation before signing. Test environments should use visibly different network identifiers and address formats to reduce accidental mainnet payments.

25. GHOST 2026 vs Zcash

The goal is not to diminish Zcash. Zcash is an important reference implementation of shielded digital cash. The table below explains how GHOST 2026 would attempt to occupy a different design position if implemented successfully.

Dimension	GHOST 2026 proposal	Zcash benchmark / contrast
Core positioning	Private digital cash plus private AI-agent settlement	Privacy-focused digital cash with mature shielded protocol.
Privacy default	Proposed shielded-by-default normal payment flow	Supports shielded and transparent capabilities; modern wallets can favor shielded use.
Proof model	Proposed setup-free modern ZK system selected after review	Modern Orchard uses Halo 2.
Consensus	Proof of Stake with deterministic finality	Zcash has historically used Equihash proof of work.
Agent delegation	Native mandates, scoped keys, budgets and sub-agent rules	Not a base-layer specialization of Zcash.
HTTP agent payments	G402 private 402 adapter	Can be integrated by applications, but not the core Zcash design goal.
MCP/AP2/A2A	First-class adapters proposed	Application-layer integration possible externally.
Network privacy	Dandelion-style propagation plus Tor/I2P targets in core design	Zcash ecosystem is actively improving Tor/network privacy; architecture differs.
Stable/private assets	Proposed shielded multi-asset note layer	ZEC is the native asset; protocol scope differs.
Selective disclosure	Incoming/outgoing viewing and purpose-limited proofs	Zcash supports viewing-key concepts; exact semantics differ.
PQ strategy	Explicit staged migration using standardized PQ primitives where appropriate	Zcash also evolves cryptography; no claim that either is currently fully post-quantum.

25.1 Where GHOST 2026 must earn credibility

A whitepaper cannot claim superiority over a network that has years of real-world adversarial operation. To become a credible Zcash alternative, GHOST 2026 would need high-quality open-source cryptography, independent audits, multi-implementation consensus testing, a substantial shielded anonymity set, reliable wallets, transparent governance and years of incident-free operation. The correct initial claim is “designed to compete,” not “proven more private.”

26. GHOST 2026 vs Historical GHOST

Area	Historical lineage	2026 proposal
Consensus	Proof of Stake	Retained, modernized for deterministic finality.
Sender privacy	Ring-signature / RingCT family	Shielded note + zero-knowledge spend proof.
Receiver privacy	Stealth-address techniques	Receiver hidden inside shielded note destination.
Amount privacy	Confidential transactions and Bulletproofs	Hidden amount inside full validity proof.
Network privacy	Dandelion++	Retained, expanded with anonymity-network

		integration targets.
Cold staking	Key-separated staking	Retained in spirit with dedicated stake authorization.
Veterans	Participation/reward concept	Optional validator/governance tier subject to decentralization limits.
Browser/app ecosystem	MetaGhost/Telaghost-era ideas	Agent wallet SDK, G402, MCP/AP2/A2A integration.
Quantum discussion	General resistance concepts	Staged standards-based PQ migration plan.

27. Security Analysis and Attack Scenarios

27.1 Prompt-injection purchase attack

Scenario: a malicious webpage instructs an autonomous agent to ignore prior instructions and pay a high-value address. Mitigation: the model never owns unrestricted signing authority. The wallet validates a structured quote against the signed mandate. If the merchant is outside scope or the amount exceeds the cap, the purchase cannot be signed even if the model insists.

27.2 Resource substitution

Scenario: a merchant obtains a valid payment credential for a cheap endpoint and reuses it to claim payment for a more expensive resource. Mitigation: the payment intent and private receipt are bound to a canonical resource digest, quote nonce, price, recipient and expiry. The credential is invalid for any different resource digest.

27.3 Replay and double charging

Scenario: the same agent authorization is submitted multiple times. Mitigation: every payment intent includes a unique nonce and the wallet records consumed authorization identifiers. Base-layer nullifiers prevent spending the same note twice; channel and voucher protocols require their own monotonic sequence or unique voucher set.

27.4 Malicious facilitator

Scenario: a G402 facilitator lies about settlement. Mitigation: facilitators do not receive custody by default and merchants can verify chain inclusion, channel state or cryptographic settlement receipts independently. Critical implementations should support multiple facilitators or direct verification.

27.5 Privacy-set fragmentation

Scenario: users split across transparent, RingCT and shielded modes, reducing effective anonymity. Mitigation: a successful 2026 upgrade should converge ordinary payments into one dominant shielded pool and make migration tooling easy. Legacy modes should be deprecated on a published schedule once ecosystem readiness permits.

27.6 Metadata deanonymization

Scenario: an observer cannot read the transaction but correlates first-seen network timing and provider logs. Mitigation: Dandelion-style relay, anonymity-network routing, standardized fee classes, batching and application-level privacy. Limit: a global observer or a merchant collecting external identity can still correlate activity.

27.7 Counterfeit-inflation bug

Scenario: an error in the zero-knowledge circuit allows creation of hidden value. Because amounts are private, detection can be difficult. Mitigation: conservation constraints must be simple, formally specified and independently audited; total shielded pool value should be tracked through auditable commitments where possible; network upgrades should include reproducible circuit builds and multiple verifier implementations. Privacy does not justify opaque monetary accounting.

27.8 Validator censorship

Scenario: validators block selected transactions or services. Mitigation: distributed validator set, slashing for provable consensus faults, multiple relay paths, fee standardization and censorship monitoring. Privacy itself helps because validators cannot easily discriminate based on hidden counterparties, though they can still censor by observable transaction identifiers or network source.

27.9 Agent key theft

Scenario: malware steals an agent capability key. Mitigation: short-lived keys, narrow mandates, rate limits, merchant restrictions, hardware-backed signing where possible, revocation and treasury separation. Loss of a capability key should not expose the root spend key.

27.10 Bridge compromise

Scenario: a bridge issuing a wrapped shielded asset is hacked. Mitigation: bridges are treated as independent risk domains. The base protocol must not equate privacy with solvency. Wallets should display the issuer or bridge trust model and caps. Native GHOST does not depend on bridge solvency.

28. Implementation Roadmap

Phase 0 - Specification and community ratification

- Publish this design as a discussion draft and separate historical claims from new proposals.
- Create an improvement-proposal repository and threat-model working group.
- Decide whether the path is an in-place GHOST upgrade, parallel pool, or clean-slate network.
- Confirm trademark, repository and governance authority before representing the project as official.

Phase 1 - Cryptographic prototype

- Implement note commitments, nullifiers and a minimal two-in/two-out shielded transfer circuit.
- Benchmark at least two candidate proving systems on desktop and mobile hardware.
- Publish test vectors and property-based tests for value conservation and serialization.
- Run independent cryptographic review before introducing real economic value.

Phase 2 - Consensus and wallet testnet

- Implement proof-of-stake finality and validator slashing in a public testnet.
- Create a reference full node, light wallet protocol and mobile wallet prototype.
- Integrate private transaction relay and Tor/I2P test modes.
- Exercise chain reorganization, validator churn and denial-of-service scenarios.

Phase 3 - Agent payment testnet

- Implement Agent Mandates and policy-engine reference library.
- Publish G402 challenge/credential specification.
- Create paid MCP and generic HTTP examples.
- Add local private receipts and human-not-present authorization flows.
- Conduct red-team prompt-injection and payment-substitution testing.

Phase 4 - Shielded assets, channels and enterprise disclosure

- Add multi-asset note support only after native shielded payments are stable.
- Prototype bilateral private channels and voucher batch settlement.
- Add selective-disclosure proofs and accountant/auditor tooling.
- Publish institutional privacy guidelines and data-minimization reference policies.

Phase 5 - Mainnet candidate

- Multiple independent security audits covering cryptography, consensus, wallet policy and networking.
- Bug bounty and public adversarial test period.

- Reproducible deterministic builds for consensus-critical components.
- Hardware-wallet and recovery validation.
- Public activation checklist with explicit unresolved risks.

Phase 6 - Post-quantum and advanced privacy research

After core stability, begin hybrid PQ keys, proof-system research, privacy-preserving reputation, recursive aggregation and more advanced channel designs. Quantum marketing should follow implementation and audit, not precede it.

29. Open Research Questions

- Which proof system offers the best combination of no trusted setup, mobile proving speed, proof size, recursion, auditability and future PQ migration?
- Can shielded staking prove bonded voting power without creating a new linkable public identity or weakening slashing?
- What receiver-scan hints reduce mobile cost without leaking durable recipient fingerprints?
- How should fee buckets be designed so they price resources accurately while minimizing unique transaction fingerprints?
- What is the safest mapping between AP2-style intent mandates and blockchain-level GHOST Agent Mandates?
- Can a merchant verify policy-compliant autonomous payment without learning the owner identity or complete mandate?
- How should sub-agent delegation be represented so every child authority is provably narrower than its parent?
- Can batch settlement preserve resource-level non-repudiation without publishing a detailed service-consumption graph?
- What Tor/I2P/Dandelion combination provides the best real-world metadata resistance for mobile wallets?
- How can private stable assets avoid turning the issuer into a global transaction metadata oracle?
- What circuit or commitment structure provides supply auditability strong enough to detect hidden inflation without exposing individual balances?
- Which portions of the stack can adopt ML-KEM/ML-DSA early, and which require entirely new ZK designs?
- How should hardware wallets display agent mandates so a human can understand the true scope before signing?
- What privacy-preserving abuse controls are possible for public paid APIs without requiring universal identity disclosure?
- How can the network maintain a large anonymity set if regulated venues support only limited receiver types?

30. Conclusion

The original GHOST project was born from a clear belief: digital cash should preserve privacy rather than create an immutable behavioral dossier. The 2026 environment makes that belief more important, not less. Autonomous agents will make far more economic decisions than humans can manually approve. If those decisions settle on transparent ledgers, the resulting graph can expose strategies, relationships, suppliers, data sources and behavior at machine speed.

GHOST 2026 therefore reframes the project around two inseparable rights: the right to transact privately and the right to delegate economic action safely. Its proposed shielded note system targets Zcash-class transaction confidentiality; its proof-of-stake and network-privacy layers preserve core GHOST lineage; and its agent mandates, G402 adapter, private receipts and machine-payment interoperability attempt to make privacy usable in the emerging autonomous economy.

The design should be judged by implementation quality, not branding. A credible privacy network requires conservative cryptography, independent review, explicit limitations, usable wallets and a community willing to reject shortcuts. If built to those standards, GHOST could become more than a historical privacy coin. It could become a private settlement layer for humans and the software agents increasingly acting on their behalf.

FINAL DESIGN PRINCIPLE

Privacy by default. Delegation by explicit mandate. Disclosure by consent. Autonomy within limits. Cryptography before marketing.

Appendix A. Conceptual Data Structures

A.1 Shielded note

```
ShieldedNote {
  version
  asset_id
  value
  recipient_diversifier
  recipient_key_component
  rho
  randomness
  memo_key_material
}

commitment = Commit(Serialize(ShieldedNote))
```

A.2 Spend description

```
ShieldedSpend {
  anchor_root
  nullifier
  proof_component
  randomized_auth_key
  spend_auth_signature
}

The consumed note and amount remain private.
```

A.3 Agent mandate

```
AgentMandate {
  mandate_id
  owner_key
  agent_key
  asset_scope[]
  max_per_payment
  max_per_period
  period_definition
  merchant_rules[]
  service_rules[]
  valid_from
  valid_until
  delegation_depth
  receipt_requirement
  revocation_nonce
  network_domain
  owner_signature
}
```

A.4 Purchase intent

```
PurchaseIntent {
  mandate_id
  merchant_credential_digest
  resource_digest
  quote_digest
  amount_commitment
  asset_id
  server_nonce
  client_nonce
  expiry
  fulfillment_terms_digest
}
```

A.5 Private receipt

```
PrivateReceipt {
  payment_commitment
  resource_digest
  merchant_digest
  fulfillment_digest
  settlement_reference
  timestamp_window
  encrypted_metadata
  merchant_signature
}
```

A disclosure proof may reveal only selected fields.

Appendix B. Example Agent-Payment Flows

B.1 Research agent buys a private data query

Owner mandate: up to 10 GHOST per day, maximum 0.05 GHOST per request, approved category “market-data”, no sub-delegation, receipt required. The research agent asks a provider for a historical dataset. The provider returns a G402 challenge containing the resource digest, 0.03 GHOST price, nonce and expiry. The policy engine confirms the category, amount and daily budget. It creates a shielded payment bound to the challenge. The provider sees evidence that its invoice was satisfied but does not see the agent's total wallet balance or other data suppliers. The local audit log records the dataset digest and private receipt.

B.2 Coding agent pays for an MCP tool

A coding agent calls a premium code-analysis tool exposed over MCP. The tool returns a payment requirement for one scan. The wallet verifies that the owner mandate allows software-development tools and caps each call at 0.02 GHOST. The G402 adapter produces a payment credential. The tool executes. For repeated scans, the agent opens a capped channel worth 1 GHOST so that each tool call updates an off-chain balance rather than settling individually.

B.3 Shopping agent with human-not-present authority

A user authorizes an agent to purchase replacement printer supplies from merchants holding an approved business credential, up to 120 units of a shielded stable asset, valid for seven days. The user is not present when the agent finds a low-stock event. The mandate permits autonomous execution. The receipt binds the merchant, item SKU digest, price and delivery terms. An accountant later receives a disclosure proof showing that the expense fell within the approved office-supplies category without seeing the user's unrelated private purchases.

B.4 Robot buys charging

A delivery robot receives a daily mobility budget. At a charging station it negotiates price per kilowatt-hour. The wallet opens or reuses a payment channel and streams signed balance updates as energy is delivered. The final channel settlement is

shielded. The station sees the robot session it physically services, but the public blockchain does not reveal the robot's other routes or charging partners.

B.5 Multi-agent procurement

A procurement orchestrator delegates 500 units to a research sub-agent and 2,000 units to a purchasing sub-agent. The research agent may buy data only; it cannot place orders. The purchasing agent may pay only merchants appearing in the approved supplier credential set. Each child mandate is narrower than the parent. Even if the research agent is compromised, it cannot use its key to transfer the full 2,500-unit budget or pay an arbitrary wallet.

Appendix C. Glossary

Agent: Software that can plan or act with delegated authority.

Agent capability key: A constrained signing key usable only within an active mandate.

AP2: Agent Payments Protocol, an open framework for agent-led payment authorization and evidence.

Commitment: A cryptographic value that binds hidden data without revealing it.

Commitment tree: Authenticated tree containing shielded note commitments.

Dandelion-style relay: A transaction propagation strategy that delays broad diffusion to reduce first-source correlation.

Deterministic finality: A consensus property under which a finalized block is not expected to be reorganized absent a severe safety violation.

G402: Proposed GHOST adapter for HTTP 402 machine payments.

GHOST Agent Mandate: Signed policy that defines an agent's allowed economic authority.

Halo 2: A modern zero-knowledge proving system used by Zcash Orchard and referenced here as a benchmark.

HTTP 402: The Payment Required HTTP status code increasingly used for machine-native payment workflows.

Incoming viewing key: Capability to discover selected incoming shielded payments without spending them.

MCP: Model Context Protocol, used by AI systems to access tools and resources.

MPP: Machine Payments Protocol, an HTTP-oriented machine-payment specification.

Nullifier: Public unique marker proving a private note has been spent once.

Orchard: Modern Zcash shielded payment protocol using Halo 2.

Proof of Stake: Consensus family in which block participation is weighted by bonded economic stake.

Private receipt: Cryptographically bound evidence of a shielded payment and resource fulfillment, encrypted or selectively disclosable.

Resource digest: Hash committing to the exact API endpoint, item, parameters or service being purchased.

RingCT: Ring Confidential Transactions, used by the historical GHOST lineage as a privacy mechanism.

Selective disclosure: Revealing only specific otherwise-private transaction information to a chosen party.

Shielded asset: An asset represented inside the private note system.

Shielded pool: Set of private notes and associated commitments/nullifiers governed by a zero-knowledge spend protocol.

Spend mandate: Delegation defining who or what may initiate spending and under what conditions.

Stealth address: Technique for deriving one-time receiver destinations, prominent in the historical GHOST design.

Viewing key: Non-spending key that reveals defined private transaction information.

x402: Open HTTP-402-based payment standard used for machine and agent payments.

Zero-knowledge proof: Proof that a statement is true without revealing the private witness used to prove it.

Zcash: Privacy-focused cryptocurrency used in this paper as a benchmark for shielded digital cash.

Appendix D. References and Source Notes

The references below are used for historical lineage, comparative design and the 2026 agent-payment environment. GHOST 2026 is an original synthesis and proposal; it does not reproduce the source whitepapers. URLs are included for verification and future research.

[1] Historical GHOST whitepaper (2020 revised edition)

Ghost - A privacy coin by John McAfee; surviving PDF mirror:

https://s2.tokeninsight.com/static/content/whitepaper/Ghost_by_McAfee_Whitepaper.727767db.pdf

[2] GHOST Privacy Whitepaper (community update, 2024)

Historical project PDF mirror previously published at:

https://ipfs.ghostbyjohnmcafee.com/static/media/Ghost_Privacy_Whitepaper.299d1822c57ee8574cbe.pdf

[3] Zcash Protocol Specification

<https://zips.z.cash/protocol/protocol.pdf>

[4] Zcash Orchard Book - keys and addresses

<https://zcash.github.io/orchard/design/keys.html>

[5] x402 - open standard for internet-native payments

<https://x402.org/>

[6] x402 technical report / whitepaper

<https://x402.org/reports/>

[7] Cloudflare Agents - x402

<https://developers.cloudflare.com/agents/tools/payments/x402/>

[8] Cloudflare Agents - Machine Payments Protocol

<https://developers.cloudflare.com/agents/tools/payments/mpp/>

[9] Google Cloud - Agent Payments Protocol (AP2)

<https://cloud.google.com/blog/products/ai-machine-learning/announcing-agents-to-payments-ap2-protocol>

[10] AP2 specification repository

<https://github.com/google-agentic-commerce/AP2>

[11] NIST FIPS 203 - ML-KEM

<https://csrc.nist.gov/pubs/fips/203/final>

[12] NIST announcement of FIPS 203, 204 and 205

<https://csrc.nist.gov/News/2024/postquantum-cryptography-fips-approved>

[13] x402 V2 overview

<https://x402.org/x402-v2-launch/>

[14] x402 batch settlement

<https://x402.org/x402-batch-settlement/>

[15] Zcash Foundation 2026 engineering/privacy direction

<https://forum.zcashcommunity.com/t/zcash-foundation-2026-stewardship-and-innovation/54477>

Source-use note

The 2020 and 2024 GHOST materials are used only to establish lineage and retained design themes. The shielded ZK ledger, AI-agent mandates, G402 payment adapter, private receipt model, delegated sub-agent policy, interoperability architecture, roadmap and security analysis in this whitepaper are newly authored for the 2026 proposal. The Zcash, x402, AP2, MPP and NIST materials are referenced to describe public technical approaches and current standards, not to imply affiliation or endorsement.

Trademark and endorsement note

GHOST, Zcash, x402, AP2, MCP, MPP and other names referenced may be trademarks or identifiers of their respective projects or organizations. This independent research draft does not claim endorsement by those projects. John McAfee died in 2021; nothing in this 2026 paper should be represented as a statement, signature or approval by him.

GHOST 2026

Privacy by default. Delegation by explicit mandate. Disclosure by consent.

End of conceptual whitepaper - Version 0.9 - October 2026