ACE Chain Technical Whitepaper

Redefining Blockchain Identity from Cryptographic Primitives

ACE Labs

Contact: jason@yeah.app

March 2026

Disclaimer: This whitepaper is provided solely for technical reference and academic discussion. It does not constitute investment advice, a securities offering, or a solicitation to purchase in any form. The technical architecture, performance projections, and cost estimates described herein are based on current designs and modeling; actual implementations may differ from these descriptions. Readers should independently assess all associated risks. ACE Labs assumes no liability for any losses arising from the use of information contained in this document.

Abstract

The blockchain industry is built upon a deeply entrenched assumption—public key equals identity. This assumption locks security, usability, and recoverability into an impossible trilemma, while creating systemic bottlenecks across critical dimensions including performance, cross-chain interoperability, and post-quantum migration.

ACE Chain begins from a single cryptographic primitive—ACE-GF (Atomic Cryptographic Entity Generative Framework)—to decouple on-chain identity from public keys. On ACE Chain, a user's on-chain identity is no longer a public key or its hash, but a zero-knowledge commitment (identity commitment, `idcom`). Public keys, signature algorithms, and even the keys themselves are reduced to replaceable authorization instruments.

This design decision unlocks a series of problems previously regarded as requiring independent solutions: sub-second cryptographic hard finality, $O(1)$ block verification, multi-chain native execution, zero-coordination state sharding, serverless wallet recovery, human-readable unified payment addressing, and high-frequency micro-settlement infrastructure for the Agent economy.

This whitepaper proceeds from the cryptographic foundations of ACE-GF, deriving the complete architecture of ACE Chain layer by layer, and demonstrating how a single root-level change restructures every critical pathway of a blockchain.

Contents

1 Project Positioning and Design Goals

ACE Chain is a Layer 1 public blockchain whose architecture is designed from cryptographic primitives upward. Its core methodology is not incremental optimization of existing blockchain systems, but rather a unified solution framework for six systemic challenges facing the industry today—derived from a single fundamental design decision: **identity–authorization separation**.

1.1 The Private-Key Trilemma

The current crypto-asset industry faces a structural contradiction involving three core requirements:

- **Self-Custody**: The user maintains complete and exclusive control over assets without reliance on any third-party intermediary.
- **Inheritance**: When a holder dies unexpectedly or becomes incapacitated, a designated beneficiary can securely assume control of the assets.
- **Yield Generation**: Assets can participate in DeFi protocols to generate returns rather than remaining idle indefinitely.

Under the existing architecture, at most two of the three requirements above can be simultaneously satisfied:

Approach	**Self-Custody**	**Inheritance**	**Yield**	**Trade-off**
Hardware wallet + DeFi	✓	×	✓	Private key becomes irrecoverable upon holder’s death; assets permanently lost
Exchange/Custodial	×	✓	✓	Asset control ceded to third party (cf. FTX collapse)
Multisig + inheritance contract	✓	✓	×	Multisig wallets are difficult to integrate with standardized DeFi vault protocols

Table 1: The pick-two-of-three constraint under traditional approaches

The root cause of this contradiction lies in the architectural assumption that "public key = identity": the private key is the sole credential for asset control, and the holder's death implies its permanent loss; any scheme that allows others to obtain the private key fundamentally compromises the security of self-custody. By industry estimates, millions of bitcoins have been permanently lost due to the unexpected death of their holders—this is not an edge case, but an inevitable consequence of the current architectural model.

ACE's solution: The ACE-GF identity–authorization separation architecture (Section 3) decouples on-chain identity from private keys, enabling the creation of dormant inheritance paths without exposing key material (CT-DAP, Section 5). Simultaneously, assets continue to generate yield within the Yault ERC-4626 smart vault (Section 6). All three requirements coexist naturally within a single identity framework.

1.2 The Signature Verification Performance Bottleneck

In current mainstream Layer 1 architectures, every transaction must carry a full digital signature (Ed25519: 64 bytes; secp256k1: 65 bytes) along with the signer's public key (32–33 bytes). Validator nodes must perform per-transaction signature verification—a computationally intensive operation (Ed25519 single verification takes approximately 76 μs). Taking Solana's Firedancer architecture as an example, each SigVerify tile processes approximately 20,000–40,000 TPS, making signature verification the **primary performance bottleneck** across all mainstream Layer 1 chains.

ACE's solution: Identity–authorization separation enables credentials to be batch-verified within zero-knowledge circuits. A single recursive proof covers all transactions in an entire block—verification cost is $O(1)$, independent of the number of transactions in the block. By eliminating the signature verification bottleneck, cryptographic hard finality time is projected to compress from Solana's approximately 12 seconds to a design target of approximately 600 milliseconds (Section 8).

1.3 Addressing and Recovery Usability Deficiencies

Blockchain addresses in the format `0x7a3b...4f2c` are not human-readable, cannot be communicated verbally, and are difficult to verify visually. Sending a transfer to another person requires copying and pasting a 42-character hexadecimal string; recovering a wallet after changing devices requires retrieving a paper backup of the mnemonic phrase and entering 12 to 24 English words one by one.

On the surface, **payment addressing** and **wallet recovery** appear to be two independent problems. In reality, they are two directions of the same underlying requirement: the former solves "how others locate me on-chain," and the latter solves "how I relocate myself on-chain."

ACE's solution: VA-DAR (Vendor-Agnostic Deterministic Artifact Resolution), an on-chain discovery registry, enables serverless wallet recovery—users can restore their complete cryptographic identity using only a password and a human-memorable identifier (such as an email address). HFI-Pay (Human-Friendly Identifier Payment), a unified payment addressing system, reduces the cognitive overhead of transfers to that of sending an email. Both systems share the same cryptographic infrastructure; a single registration simultaneously provides recovery capability and payment reachability (Section 4). This fundamentally eliminates the technical-specialist barrier that has long confined crypto assets to a niche audience, paving the way for cryptocurrency to enter the daily lives of ordinary people.

1.4 Quantum Computing Threat and Migration Cost

When Shor's algorithm gains the capability to break elliptic-curve cryptography, every blockchain built on the "public key = identity" assumption will face the same catastrophic scenario: **replacing the signature algorithm is equivalent to replacing every user's identity**. Each on-chain account requires state migration, every smart contract referencing legacy addresses needs index updates, and legacy keys remain exposed to quantum attack risk throughout the migration window.

ACE's solution: The on-chain identity is `idcom`—a cryptographic commitment that reveals no public-key information—rather than a public key itself. The signature algorithm serves as a replaceable authorization instrument (the Tagged Signature mechanism supports Ed25519, secp256k1, ML-DSA-44, HMAC-SHA256, and others), reducing the quantum threat to a simple parameter switch—changing the algorithm does not affect account identity, achieving zero-migration-cost transition (Section 3 § 3.6, Section 12 § 12.2).

1.5 Cross-Chain Identity Fragmentation

The same user holds entirely different addresses on Ethereum (secp256k1), Solana (Ed25519), and Bitcoin (secp256k1 + Script), derived from entirely different key pairs. No purely cryptographic method exists to prove that these addresses belong to the same natural person—because at the cryptographic level, they are indeed entirely unrelated. Cross-chain identity aggregation currently relies solely on centralized third-party attestations.

ACE's solution: ACE-GF starts from a single REV (Root Entropy Value) and deterministically derives independent key streams for seven chains via HKDF-SHA256 domain separation (Solana, Ethereum, Bitcoin, Polkadot, Cosmos, X25519 end-to-end encryption, ML-DSA-44 post-quantum signatures), with all keys sharing a common cryptographic root. The n-VM multi-chain scheduler natively executes EVM/SVM/BVM/TVM transactions within the same state tree, eliminating the need for cross-chain bridges (Section 3 § 3.3, Section 9 § 9.1).

1.6 Finality Efficiency

Mainstream Layer 1 chains, exemplified by Solana, require 31 block confirmations (approximately 12 seconds) for hard finality. Furthermore, BFT votes are submitted as on-chain transactions, consuming approximately 65% of total TPS, with validators network-wide paying approximately

$67.5 million annually in voting fees. This finality mechanism is fundamentally probabilistic (based on economic security assumptions) rather than cryptographically deterministic.

ACE's solution: Zero-knowledge proofs provide cryptographic hard finality for each block (target approximately 600 ms). BFT votes are transmitted as off-chain messages, generating no on-chain transactions and completely eliminating voting costs. Modeling analysis projects that the aggregate cost per million transactions drops from Solana's approximately $2.86 to approximately $0.01 (Section 8, Section 11). These figures represent design targets based on structural analysis, not production measurements.

1.7 Infrastructure Gap for the Agent Economy

The AI Agent economy is emerging rapidly: autonomous agents execute payments, procurement, negotiation, and asset management on behalf of humans. This paradigm imposes entirely new performance and cost requirements on the underlying blockchain:

- **High-frequency micro-transactions**: Negotiation, quoting, and settlement between agents may generate dozens of transactions per second, each of minuscule value (cent-level). The fee structures of current mainstream chains (Solana: $0.00025 per transaction; Ethereum: $0.50–$50 per transaction) render the majority of micro-transaction scenarios economically infeasible.
- **Deterministic finality**: Agents cannot "wait and confirm later" the way humans can. Automated workflows require transactions to achieve irreversible finality at sub-second latency; otherwise, complex compensation logic must be introduced.
- **Policy-bounded execution**: Agents operating assets on behalf of humans must execute within preset policy boundaries (limits, whitelists, time windows), with immediate freezing and escalation to human review upon policy violations.
- **Zero-fee DeFi within the runtime**: When DeFi operations (swaps, lending, yield farming) execute within the same runtime, they should incur no additional cross-contract invocation overhead. The Agent economy requires DeFi to function as infrastructure, not as a profit center.

ACE's solution: ACE Chain's sub-second hard finality and near-zero transaction cost (approximately $0.01 per million transactions) provide an economically viable settlement layer for the Agent economy. The AESP (Agent Economic Sovereignty Protocol) SDK delivers a policy engine, identity derivation, negotiation state machine, and human review queue, enabling agents to autonomously execute economic actions under human sovereignty constraints. The Yault vault provides DeFi primitives within the runtime at near-zero fees (Section 7, Section 11).

1.8 Design Goals Summary

Industry Pain Point	Design Goal	A So lu ti Pa
Private-key trilemma	Simultaneous self-custody, inheritance, and yield	AC GI DA
Signature verification bottleneck	Eliminate per-transaction signature verification overhead	*O* re- cu siv ZK pr
Addressing and recovery deficiencies	Human-readable payment and recovery	VA DA Pa
Quantum computing threat	Zero-migration-cost algorithm switching	`id` co mı mo Si na tu
Cross-chain identity fragmentation	Unified identity with multi-chain native execution	7- str HI VI
Finality efficiency	Sub-second cryptographic hard finality	ZK fi- na ity ce tif ca
Agent economy infrastructure	High-frequency micro-payments, policy boundaries, zero-fee DeFi	AI ru tin De

Table 2: ACE Chain design goals and solution paths

All of the above solution paths share a single cryptographic premise: **identity–authorization separation**. The subsequent sections derive ACE Chain's complete technical architecture from this design decision, layer by layer.

2 Core Insight: Identity–Authorization Separation

2.1 Breaking the Fundamental Equation

The core design decision of ACE Chain is to break the fundamental equation that the blockchain industry has implicitly followed since the inception of Bitcoin:

$$\text{Public Key} = \text{Identity} = \text{Address} \tag{1}$$

ACE Chain decomposes this trinity into two independent abstraction layers:

```
Identity layer: idcom = Commitment(REV, salt, domain) // On-chain identity
Authorization layer: credential = f(attest_key, payload) // Replaceable credential
```

- **idcom (identity commitment)**: A 32-byte cryptographic commitment that serves as the user's on-chain identity. It reveals no public-key information and is not bound to any specific signature algorithm.
- **credential**: An authorization credential proving that the transaction originator is entitled to act on behalf of this `idcom`. Its concrete form may be an Ed25519 signature, a secp256k1 signature, an HMAC-SHA256 message authentication code, or even a post-quantum signature—the identity remains unchanged while authorization instruments can be independently replaced.

2.2 Multiple Corollaries from a Single Decision

Identity–authorization separation is not an isolated design choice; it is a **generative principle**: from this principle, a series of problems previously regarded as requiring independent solutions are naturally transformed into its corollaries.

Corollary	**Mechanism**
$O(1)$ block verification	Credentials verified in ZK circuits; recursive proof covers entire block
Sub-second hard finality	ZK proofs provide cryptographically deterministic finality
Zero-cost PQC migration	Identity is `idcom`, not a public key
Multi-chain native execution	Same `idcom` derives per-chain keys via HKDF streams
Zero-coordination sharding	HKDF context isolation; same identity, different shards, cryptographically inc
Serverless recovery	REV deterministically reconstructed from password and identifier
Human-readable addressing	DiscoveryID maps human identifiers to on-chain identity

The cryptographic prerequisite underlying all of the above corollaries is ACE-GF—the cryptographic framework that makes identity–authorization separation technically feasible.

3 ACE-GF: The Cryptographic Foundation

ACE-GF is not a bespoke component tailored specifically for ACE Chain. The reality is precisely the reverse: the entire architectural vision of ACE Chain originates from the design space opened up by the ACE-GF cryptographic framework.

3.1 Design Philosophy

ACE-GF (Atomic Cryptographic Entity Generative Framework) is a cryptographic identity generation framework whose core proposition is: **starting from a memorized secret, deterministically generate a complete multi-chain cryptographic identity without persisting any key material.**

```
User's memorized secret (password + identifier)
       |
   Deterministic derivation
       |
   +------------------------------+
   | 7-chain signing keys |
   | Identity commitment (idcom) |
   | Attestation key (attest_key)|
   | Encryption key (X25519) |
   | Post-quantum key (ML-DSA-44)|
   | Recovery address (DiscoveryID)|
   +------------------------------+
```

3.2 REV: Root Entropy Value

The starting point of ACE-GF is the REV (Root Entropy Value)—a 32-byte root identity material. The core security property of the REV is **non-persistence**: it is reconstructed from the user's password when needed and securely erased (zeroized) from memory immediately after use.

Two paths exist for obtaining the REV:

Path A: Initial generation.

```
Random entropy $\to$ BIP39 mnemonic $\to$ encoded as REV32 (32-byte canonical format
   )
```

Path B: Recovery from Sealed Artifact.

```
SA (Sealed Artifact) + user password
```

```
    |
K_base = Argon2id(password, "ACEGF-KDF-GLOBAL-V1")
    |
K_sealed = HKDF(K_base, "ACEGF-KDF-NATIVE-V1")
    |
REV = AES-256-GCM-SIV-Decrypt(SA, K_sealed)
```

The SA (Sealed Artifact) is a 32-byte encrypted container that can be stored at any location (cloud storage, IPFS, on-chain registry, etc.). Without the correct password, the SA is computationally indistinguishable from a random byte sequence.

3.3 Seven-Stream HKDF Derivation

Starting from the REV, ACE-GF deterministically derives seven cryptographically independent key streams via HKDF-SHA256 domain separation:

Stream	Domain Tag	Algorithm	Purpose
1	`ACEGF-V1-ED25519-SOLANA`	Ed25519	Solana signing
2	`ACEGF-V1-ED25519-POLKADOT`	Ed25519	Polkadot signing
3	`ACEGF-V1-SECP256K1-EVM`	secp256k1	Ethereum/EVM signing
4	`ACEGF-V1-SECP256K1-BTC`	secp256k1	Bitcoin Taproot
5	`ACEGF-V1-SECP256K1-COSMOS`	secp256k1	Cosmos signing
6	`ACEGF-V1-X25519-IDENTITY`	X25519	End-to-end encryption
7	`ACEGF-V1-ML-DSA-44-PQC-IDENTITY`	ML-DSA-44	Post-quantum signing

Table 4: ACE-GF seven-stream key derivation

Each stream's seed is guaranteed to be cryptographically independent through a distinct `info` tag. To prevent domain-collision attacks (where different `info` and `context` pairs could collide under simple concatenation), ACE-GF employs a **length-prefixed derivation** (`hkdf_expand_with_context`):

$$\text{info}[i] = \text{LE32}(\text{domain\_tag}[i].\text{len}()) \parallel \text{domain\_tag}[i] \parallel \text{context} \tag{2}$$

Where `LE32` is a 4-byte little-endian length prefix. This ensures that the derivation remains unambiguous across all context configurations.

The security guarantees of HKDF (based on the pseudorandom function assumption of the underlying HMAC) ensure that keys produced by different domain tags are computationally unlinkable. Compromising any single stream does not affect the security of the remaining six.

3.4 Identity Commitment: idcom

`idcom` is the on-chain identity on ACE Chain, constructed as follows:

```
identity_root = HKDF(K_master, "acegf:identity:root")
idcom = SHA-256(identity_root || context || domain)
```

Where:

- `identity_root`: Identity root material derived from the master key.
- `context`: An optional context tag for shard isolation or multi-vault support.
- `domain`: Concatenation of chain ID and slot number (providing replay protection).

`idcom` possesses the following key properties:

- **Hiding**: Reveals no public key or any key material.
- **Determinism**: The same REV + context + domain always produces the same `idcom`.
- **Context isolation**: Different context values produce cryptographically independent `idcom` values.
- **Algorithm agnosticism**: `idcom` is not bound to any signature algorithm; changing the signature scheme does not affect on-chain identity.

3.5 Context Isolation and Multi-Vault Support

ACE-GF achieves cryptographic-grade state isolation through the `context` parameter of HKDF:

$$\mathrm{HKDF}(\mathrm{REV}, \mathrm{info}, \mathrm{context} = \text{“treasury:0”}) \neq \mathrm{HKDF}(\mathrm{REV}, \mathrm{info}, \mathrm{context} = \text{“payment:0”}) \tag{3}$$

Different `context` values produce entirely independent key sets and `idcom` values. At the application layer, this mechanism supports multi-vault management; at the protocol layer, it enables zero-coordination state sharding (Section 10).

3.6 The Cryptographic Prerequisite for Breaking the Trilemma

ACE-GF's identity–authorization separation is the **cryptographic precondition** for simultaneously achieving self-custody, inheritance, and yield generation:

Dimension	Solution	Mechanism
Self-custody	ACE-GF autonomous key derivation	Full user control; REV is never persistently stored
Inheritance	CT-DAP condition-triggered authorization	Identity $\neq$ private key; dormant inheritance paths can be created without exposing key material
Yield	Yault ERC-4626 vault	Standardized yield interface; self-custodied assets participate directly

Table 5: How ACE-GF breaks the trilemma

Key insight: **Identity is no longer equivalent to the private key.** Therefore:

- The definition of self-custody evolves from "only I can access the private key" to "only my `idcom` can authorize operations."
- Inheritance does not require exposing the private key to the beneficiary—CT-DAP creates an independent authorization path that remains unavailable until the trigger condition is met.
- Yield generation does not require transferring assets to a third party—the Yault vault operates within the same state tree, and self-custodied assets participate directly in yield generation.

All three requirements coexist naturally within the same identity–authorization separation framework, because they operate on the authorization layer (replaceable, composable) rather than the identity layer (`idcom` remains immutable throughout).

4 Human Reachability: VA-DAR Wallet Recovery and HFI Unified Payment Addressing

4.1 Problem Definition

The user experience bottleneck in current blockchain systems lies not in transaction speed, but in **addressing**.

Payment addressing and wallet recovery appear on the surface to be two independent product requirements, but they are in essence two directions of the same underlying problem:

- **Payment addressing**: How do others locate me on-chain?
- **Wallet recovery**: How do I relocate myself on-chain?

ACE Chain resolves both directions through two synergistic systems: VA-DAR handles recovery, and HFI-Pay handles payment addressing. Both share the same cryptographic infrastructure.

4.2 VA-DAR: Serverless Wallet Recovery

VA-DAR (Vendor-Agnostic Deterministic Artifact Resolution) is an on-chain discovery registry that enables users to restore their complete cryptographic identity using only a password and a human-memorable identifier (such as an email address).

Core construct: DiscoveryID.

```
K_base = Argon2id(password, "ACEGF-VADAR-V1:" || normalized_email)
K_idx = HKDF(K_base, "va-dar:discovery:index")
DiscoveryID = HMAC-SHA256(K_idx, normalized_email)
```

DiscoveryID is a 32-byte deterministic identifier with two key properties:

- **Determinism**: The same password and the same email always produce the same DiscoveryID.
- **Privacy preservation**: Only the DiscoveryID is stored on-chain, not the original identifier; reverse-engineering the email or password from the DiscoveryID is computationally infeasible.

On-chain registry structure:

```
DiscoveryID $\to$ DiscoveryRecord {
    commit: SHA-256(SA2), // Commitment to SA2
    owner_pubkey: TaggedPubkey, // Bound at first write, immutable
    sealed_artifact: Option<Vec<u8>>, // SA2 ciphertext (max 4 KiB)
    version: u64, // Monotonically increasing version
    created_at: u64, // Registration slot
}
```

First-write binding: The `owner_pubkey` of a registry entry is atomically bound at initial registration; subsequent update operations must be signed by this public key. This mechanism ensures that even if an attacker learns the user's identifier, they cannot overwrite the registration record.

Tombstone mechanism: To support identity migration or revocation (e.g., when a user changes their passphrase), VA-DAR implements a `tombstone` state. A signed tombstone marker renders the record inactive for updates while preserving its history, preventing stale registration data from being exploited during recovery.

Recovery flow:

```
User input: password + email
    |
(1) Compute DiscoveryID
    |
(2) Query on-chain registry $\to$ obtain SA2 (Sealed Artifact)
    |
(3) K_sa = HKDF(K_base, "acegf:sa2:seal")
    mnemonic = AES-256-GCM-SIV-Decrypt(SA2, K_sa, AAD=email)
    |
(4) Reconstruct REV from mnemonic $\to$ 7-stream HKDF derivation $\to$ full identity
     recovery
```

The entire recovery process depends on no third-party server. Users need only remember two things: their password and their identifier.

4.3 HFI-Pay: Human-Friendly Identifier Payment

HFI (Human-Friendly Identifier) is a payment addressing system based on human-readable identifiers. Its design goal is to reduce the cognitive overhead of on-chain transfers to the level of email: **enter the recipient's email or phone number to complete a payment**.

Three-layer architecture:

Layer	Responsibility	Chara
Protocol layer (on-chain)	Intent addressing, signature verification, state transitions	Pure mathematics, publicly auditable
Application layer (Relay)	Identifier-to-intent mapping, push notifications, gas sponsorship	Operat tier, replaceable
Identity layer (OTP verification)	Real-person identity verification	Accoun mechanism, standardized interface

Table 6: HFI-Pay three-layer architecture

Path A: Recipient already registered (Direct Deposit Optimization). When the recipient has already bound their XID (Extended Identity) in the HFI-Pay registry, the system enables an atomic `direct_deposit` optimization. The relay bypasses the intermediate deposit address and claim phase, transferring funds directly from the sender to the recipient's XID-derived account in a single step, while preserving the intent record for auditability.

```
Sender input: recipient email + amount
    |
Query registry: email $\to$ XID $\to$ AccountId
    |
Direct transfer: sender account $\to$ recipient account
```

Path B: Recipient not yet registered (intent-based payment). When the recipient has not yet registered, HFI-Pay employs an intent mechanism: funds are first deposited into a temporary escrow address deterministically derived from the `intentId`; the recipient claims the funds after verifying their identity via OTP; funds are automatically refunded to the sender if unclaimed after timeout.

Intent state machine: `Created` → `Funded` → `Claimed` → `Withdrawn`, or `Created` → `Funded` → `Expired` → `Refunded`.

4.4 Synergy Between VA-DAR and HFI-Pay

Both systems are initialized simultaneously during user registration:

```
User registration flow:
    |
(1) Generate wallet: ACE-GF $\to$ mnemonic + 7-chain keys + XID
    |
(2) VA-DAR backup: seal SA2 $\to$ compute DiscoveryID $\to$ write to on-chain
    registry
    |
(3) HFI-Pay registration: email + XID + signature $\to$ write to payment registry
    |
(4) Complete: a single operation provides both recovery capability and payment
    reachability
```

Thereafter:

- **Others locate you**: Query the HFI-Pay registry by identifier, obtain the XID, and initiate payment.
- **You locate yourself**: Password + identifier → VA-DAR → SA2 → reconstruct REV → recover full cryptographic identity.

One entry point (a human-memorable identifier), two directions (outward reachability, inward recoverability), zero server dependency.

5 CT-DAP: Crypto-Asset Inheritance

5.1 Problem Definition

Under the traditional "public key = identity" architecture, crypto-asset inheritance faces a fundamental difficulty: the private key is the sole credential for asset control, and the holder's death implies the permanent loss of that credential. All existing approaches suffer from structural deficiencies: will-and-lawyer schemes require exposing the private key to a third party, compromising self-custody; multisig schemes require the beneficiary to participate in configuration in advance and are difficult to integrate with DeFi; time-lock schemes lack flexibility and cannot accommodate complex trigger conditions.

5.2 CT-DAP Design

CT-DAP (Condition-Triggered Dormant Authorization Paths) creates a dormant authorization path on top of identity–authorization separation—a path that remains unavailable until specific conditions are triggered; once conditions are met, the designated beneficiary can securely assume control of the assets.

Core design principles:

- The asset holder is never required to expose the private key to anyone.
- The beneficiary cannot access assets until the trigger condition is satisfied.
- Once the condition is met, the beneficiary can complete the claim without the holder's participation.
- Assets continue to generate yield in the Yault vault throughout the dormancy period.

5.3 Oracle Quad-Source Verification

Chainlink CRE (Chainlink Runtime Environment) performs four independent verifications in parallel before submitting an attestation:

Data Source	Purpose
drand beacon	Cryptographic timestamp proof (whether the holder has exceeded the activity deadline)
Vault balance	On-chain call confirming the vault holds sufficient assets
Compliance API	Recipient KYC/AML screening
Price oracle	Asset price verification (preventing malicious triggers under abnormal market conditions)

Table 7: CT-DAP oracle quad-source verification

Only after all four verifications pass may the Oracle submit a RELEASE attestation. Once submitted, the RELEASE attestation possesses **release finality**—it cannot be overridden or revoked by any source.

5.4 Relationship to Identity–Authorization Separation

CT-DAP is feasible precisely because identity $\neq$ private key: the holder's `idcom` remains unchanged; CT-DAP creates an independent authorization path that involves none of the holder's key material; the beneficiary completes the claim by signing with their own keys.

6 Yault: Self-Custody Yield Vault

6.1 ERC-4626 Smart Vault

Yault is a smart vault system based on the ERC-4626 standard, providing a standardized yield interface for self-custodied assets. Core operations include `deposit` (deposit underlying assets, receive share tokens), `redeem` (redeem share tokens, withdraw underlying assets plus accumulated yield), and `transfer` (transfer share tokens between parent and sub-accounts).

6.2 Synergy with CT-DAP

The synergy between Yault and CT-DAP is the key to breaking the trilemma: once assets are deposited into the Yault vault, share tokens continuously generate yield; simultaneously, a portion of shares can be allocated to CT-DAP inheritance paths. Assets thus exist simultaneously in three states—self-custodied (user retains full control of share tokens), yield-generating (Yault continuously produces returns), and inheritable (CT-DAP path is ready, released upon condition trigger). These three states are not mutually exclusive.

6.3 Agent Spending Policies

Yault supports Agent Spending Policies, allowing holders to authorize specific agents to use vault assets within preset limits and conditions. Constraint dimensions include spending limits (daily/monthly caps), usage restrictions (only permitted operation types), and temporal constraints (authorization validity period).

7 AESP: Agent Economic Sovereignty Protocol

7.1 Design Motivation

AI agents executing economic actions on behalf of humans is an irreversible trend. However, granting agents payment capabilities introduces a fundamental tension: agents require sufficient autonomy to operate efficiently, yet humans must retain ultimate control over their assets. AESP (Agent Economic Sovereignty Protocol) addresses precisely this tension—enabling agents to efficiently execute economic actions within a framework of **bounded autonomy**.

7.2 Architecture Overview

AESP is a policy and payment protocol layer built on top of ACE Chain, with its architecture divided into five modules:

Module	Responsibility	Key Capabilities
Policy Engine	Policy evaluation and execution boundaries	8 checks: per-transaction limit, rolling budget (daily/weekly/monthly), address whitelist, chain whitelist, time window, first-payment review, minimum balance, budget tracking
Identity	Agent identity derivation	BIP44 deterministic key derivation; each agent receives an independent Ed25519 key pair
Negotiation	Inter-agent negotiation	State-machine-driven offer/counter-offer/accept/reject flow with session persistence
Commitment	Verifiable commitments	EIP-712 structured commitments, dual-party signatures, on-chain escrow settlement
Review	Human review queue	Automatic escalation to human review upon policy violation, with emergency freeze support

Table 8: AESP module architecture

7.3 Policy Engine: 8 Checks

The core of AESP is a deterministic policy engine. Every economic action initiated by an agent must pass the following 8 checks before execution:

1. **Per-transaction limit**: Transaction amount does not exceed the preset ceiling.
2. **Rolling budget**: Cumulative daily/weekly/monthly spending does not exceed respective caps.
3. **Address whitelist**: Recipient address is on the allowlist.
4. **Chain whitelist**: Target chain is on the allowlist.
5. **Time window**: Current time falls within the permitted execution window.
6. **First-payment review**: Mandatory human confirmation for first payment to a new address.

7. **Minimum balance**: Post-execution account balance does not fall below the safety threshold.
8. **Budget audit**: Complete audit log supporting post-hoc traceability.

If all checks pass, execution proceeds automatically; if any check fails, the action is escalated to the Review module's human review queue, and the agent's operation is suspended pending human decision.

7.4 Synergy with ACE Chain

The combination of AESP and ACE Chain produces distinctive architectural advantages:

- **Sub-second finality × Agent automation**: An agent's negotiation–settlement cycle can complete within a single slot (400 ms), eliminating the need for compensation logic.
- **Context sharding × Privacy isolation**: AESP's ephemeral address pool leverages ACE-GF's HKDF context isolation to generate a cryptographically independent address for each agent transaction.
- **Yault vault × Policy boundaries**: Agent spending policies map directly to Yault's Agent Spending Policies, enforced on-chain.
- **Near-zero fees × Micro-economy**: ACE Chain's cost of approximately $0.01 per million transactions makes cent-level micro-transactions economically viable.

7.5 A2A Interoperability

AESP implements the Google A2A (Agent-to-Agent) protocol's Agent Card builder, enabling agents on ACE Chain to be discovered and invoked by other agents. Exposed capabilities include payment, negotiation, data queries, commitment signing, permission delegation, and arbitration.

8 Consensus and Finality

8.1 Consensus Model

ACE Chain employs a hybrid BFT + PoH + ZK consensus model:

- **PoH (Proof-of-History)**: A serial SHA-256 hash chain provides verifiable temporal ordering without reliance on wall-clock synchronization.
- **BFT voting**: $\frac{2}{3}$ stake-weighted voting achieves soft finality (approximately 400 ms).

- **Capability Committee Certificates**: For specific execution domains (e.g., Bitcoin payments, Solana light-client verification), a specialized sub-committee generates threshold certificates (`CommitteeCertificate`) proving execution correctness without requiring full-node participation for every specialized domain opcode.
- **ZK proofs**: STARK/FRI proofs (via the Winterfell proving framework) provide cryptographic hard finality.

8.2 Two-Tier Finality Model

Tier	Latency	Mechanism	Security Guarantee
Soft finality	∼400 ms	$\frac{2}{3}$ stake-weighted BFT voting	Economic security ($\frac{1}{3}$ stake slashable)
Hard finality	Target ∼600 ms	STARK/FRI proof verification	Cryptographic certainty (mathematically unforgeable)

Table 9: ACE Chain two-tier finality model

The finality state machine comprises five states: Pending → Soft (received $\frac{2}{3}$ votes) → Hard (received valid ZK finality certificate).

Slot-based Snapshotting: To support high-frequency sub-second finality, the engine maintains deterministic `ShardedStateSnapshots` for each slot. If a builder timeout or proof failure occurs, the engine can instantaneously roll back to the last conflict-free snapshot and requeue transactions, ensuring consensus continuity.

8.3 $O(1)$ Block Verification

Identity–authorization separation makes an entirely new verification model possible.

Traditional model (e.g., Solana): Per-transaction signature verification; verification cost $O(n)$.

$$T_{\text{verify}} = n \times 76\ \mu\text{s (Ed25519 verification)} \tag{4}$$

ACE model: A single recursive ZK proof; verification cost $O(1)$.

$$T_{\text{verify}} \approx 0.5 \text{ ms (STARK verification), independent of transaction count} \tag{5}$$

The STARK circuit asynchronously proves the following compound statement on GPU: "For each transaction in this block, there exists an attest_key such that (1) `idcom` = Commitment(REV, salt, domain), (2) attest_key = HKDF(REV, info, domain), (3) credential = HMAC(attest_key, payload_hash ∥ domain), and attest_key has never been leaked."

Proof generation proceeds asynchronously on GPU (approximately 240 ms/slot) using the Winterfell STARK prover, pipelined across slots to run in parallel with block production, never blocking the critical path. The choice of STARK/FRI over Groth16 eliminates the need for a trusted setup and provides 128-bit post-quantum security for the proof system itself.

8.4 Leader Election

Deterministic leader election based on stake weight:

```
seed = SHA-256(hash of the last finalized block)
slot_hash = SHA-256(seed || slot_number)
leader = rejection_sampling(slot_hash mod total_stake $\to$ corresponding validator)
```

Each epoch's seed is determined by the last finalized block, preventing leader prediction beyond a single epoch.

9 Execution Layer: n-VM and Parallel Scheduling

9.1 n-VM Multi-Chain Scheduling

ACE Chain's execution layer is not a single virtual machine but an n-VM scheduler that routes transactions to the corresponding execution engine based on the opcode prefix of the transaction payload:

Opcode Range	**Engine**	**Functionality**
0x01–0x0F	ACE Native	Native transfers, account creation
0x10–0x1F	EVM (revm)	Ethereum smart contracts (Shanghai-compatible)
0x20–0x2F	SVM	Solana BPF programs
0x30–0x3F	BVM	Bitcoin Script + UTXO model
0x40–0x4F	TVM	Tron-compatible contracts

Table 10: n-VM execution engine routing

This is not "compatibility" achieved through bridging—native transaction formats from each chain can be submitted directly to ACE Chain for execution, with the unified state tree and consensus layer providing security guarantees.

Deterministic Address Projection: Every external VM address (e.g., 20-byte EVM/Tron, 32-byte SVM) is projected to a canonical ACE `AccountId` via a deterministic domain-prefixed hashing scheme (e.g., `SHA-256("evm:" || address)`). This ensures that a single ACE-GF identity maintains a consistent, bridge-less presence across all execution engines.

9.2 Transaction Processing Pipeline

ACE Chain employs a three-phase pipeline: Attest → Execute → Prove.

Phase 1a — Attest (CPU parallel, approximately 2–5 μs/tx): Non-empty payload check, payload binding verification, domain binding verification, identity existence verification, credential verification (Ed25519/secp256k1 signature verification; HMAC-SHA256 type deferred to Phase 2 ZK circuit verification).

Phase 1b — Execute (batch parallel, approximately 10–300 μs/tx): The n-VM scheduler routes to the corresponding engine by opcode; write-set analysis → greedy batching → intra-batch rayon parallel execution; outputs state deltas and execution receipts.

Phase 2 — Prove (GPU asynchronous, off the critical path): Per-transaction ZK-ACE proof → recursive aggregation into a block-level proof → finality certificate generation.

9.3 Parallel Scheduling Model

Write-set-based automatic parallel scheduling follows this workflow: for each transaction, extract the write-set (the set of accounts involved) based on its opcode; a greedy scheduling algorithm places it into the earliest conflict-free batch; transactions from the same sender are strictly ordered by nonce. Batches execute sequentially; within each batch, rayon achieves CPU-core-level parallelism.

EVM/TVM general contract calls, whose write sets cannot be statically determined, are tagged as global write sets (`WriteSet::Global`) and forced into serial execution. This limitation is analogous to the bottleneck Solana faces when handling cross-program invocations (CPI).

10 State Sharding: HKDF Context Isolation

10.1 Sharding Principle

The context isolation property of ACE-GF naturally supports zero-coordination state sharding:

$$\mathrm{HKDF}(\mathrm{REV}, \mathrm{info}, \mathrm{context} = \text{“shard:0”}) \neq \mathrm{HKDF}(\mathrm{REV}, \mathrm{info}, \mathrm{context} = \text{“shard:1”}) \tag{6}$$

The account address spaces under different contexts are cryptographically disjoint—no cross-shard coordination is needed to guarantee the absence of address collisions.

10.2 Shard Routing

$$\mathrm{shard\_id} = \text{SHA-256}(\mathrm{vm\_prefix\_length} \parallel \mathrm{vm\_prefix} \parallel \mathrm{context\_tag}) \mod \mathrm{NUM\_SHARDS} \tag{7}$$

The target shard for a transaction can be determined during Phase 1a (Attest), with zero additional computational overhead.

10.3 Scalability Projections

Shards	Independent TPS	Shared TPS	Total	Relative to Solana
1	∼10,000–20,000	—	∼10,000–20,000	∼2.5–5×
2	∼14,000	∼6,000	∼20,000	∼5×
4	∼28,000	∼6,000	∼34,000	∼8.5×
8	∼35,000	∼6,000	∼41,000	∼10×

Table 11: Sharding scalability projections (blueprint modeling values)

Note: The figures above are projections based on architectural modeling, not benchmark results from the current implementation. The single-shard 10,000–20,000 sustained TPS represents a directional estimate consistent with the performance analysis in Section 11.

11 Performance and Cost Analysis

11.1 Protocol-Guaranteed Performance (Tier 1)

The following metrics are directly guaranteed by ACE Chain's cryptographic architecture and consensus protocol. They are independent of engineering optimization; competitors cannot replicate these properties through increased engineering investment—doing so would require redesigning their account models and consensus protocols.

Metric	ACE Chain	Industry Status Quo	Guarantee Mechanism
Transaction finality	0.6–2.4 s (hard)	Solana ∼12 s, ETH ∼13 min	BFT voting + STARK proof verification
Block verification	$O(1)$	$O(N)$	ZK-ACE compresses N tx into 1 STARK proof
PQC migration impact	0% TPS change	Solana −90%, ETH −85%	ZK-ACE absorbs signature algorithm differences
Per-tx authorization cost	→ 0 (amortized)	ETH \$0.50–\$5, SOL \$0.00025	Fixed: 1 STARK verification per block

Table 12: Protocol-guaranteed performance metrics (Tier 1)

11.2 Post-Quantum Era Performance Impact

NIST FIPS 203/204/205 were finalized in 2024. NSM-10 and CNSA 2.0 mandate compliance by 2027–2030. At that point, all blockchains employing per-transaction signature verification will face 80–90% TPS degradation and order-of-magnitude cost increases, while ACE Chain's performance and cost structure remain unchanged:

Chain	Sig. Size Change	Sustained TPS Change	Relative Advantage
Bitcoin	64 B → 2,420 B (37.8×)	∼7 → ∼1–2 (−80%)	—
Ethereum	64 B → 2,420 B (37.8×)	∼15–30 → ∼3–5 (−85%)	—
Solana	64 B → 2,420 B (37.8×)	∼2–4K → ∼200–500 (−90%)	2–5× → 20–100×
ACE Chain	0 B → 0 B (unchanged)	10–20K → 10–20K (0%)	Unchanged

Table 13: Post-quantum era performance impact projections

11.3 Pipeline Comparison with Solana

Dimension	ACE Chain	Solana
Block time	400 ms	400 ms
Soft finality	∼400 ms ($\frac{2}{3}$ BFT voting)	∼400 ms (optimistic confirmation)
Hard finality	Target ∼600 ms (STARK/FRI proof)	∼12 s (31 confirmations)
Block verification	$O(1)$ (single recursive proof)	$O(n)$ (per-tx signature verification)
Signature verification cost	Verified in ZK circuit (GPU async)	∼76 μs per tx (CPU)
On-chain signature storage	None (credentials consumed in proof)	96 B/tx (signature + public key)
Vote transactions	Off-chain messages	On-chain transactions (∼65% of total TPS)

Table 14: ACE Chain vs. Solana pipeline comparison

11.4 Cost per Million Transactions

Based on Solana mainnet operational data (2025–2026) and the ACE Chain architectural model:

Solana: Approximately 1,500 active validators; total network annual operating cost approximately $90 million (including approximately $67.5 million in voting fees); actual user TPS approximately 1,000 (excluding vote transactions); annual transaction volume approximately 31.5 billion.

ACE Chain (single-shard model): Approximately 200 validators + 2 GPU prover nodes; validator annual cost approximately $1.44 million; GPU annual cost approximately $53,000; total network annual cost approximately $1.5 million; target sustained TPS approximately 10,000–20,000; annual transaction volume approximately 157.7–630 billion.

Metric	Solana	ACE (1 shard)	ACE (4 shards)
Annual network cost	∼$90M	∼$1.5M	∼$3M
Annual tx volume	∼31.5B	∼157.7B	∼536B
Cost per million tx	∼$2.86	∼$0.0095	∼$0.0056

Table 15: Aggregate cost per million transactions comparison

The cost advantage derives from three primary sources: (1) no on-chain vote transactions, eliminating the single largest expenditure item on Solana; (2) $O(1)$ verification means validator nodes require no GPU—only a small number of prover nodes need high-end hardware; (3) GPU proving cost is amortized per slot, with a single H100 computation (approximately 240 ms) covering all transactions in the entire slot.

Caveats: Solana's cost data is based on actual mainnet operations; ACE's data is based on architectural modeling. The operational maturity gap may translate into additional hidden costs during the project's early stages.

11.5 Theoretical Peak TPS vs. Sustained Throughput

Distinguishing peak TPS (theoretical upper bound) from sustained TPS (steady-state throughput) is essential for accurately evaluating chain performance. Industry experience indicates that sustained throughput is typically 3–10% of peak.

Metric	**ACE Chain**	**Solana**	**Notes**
Theoretical peak TPS (in-memory)	∼170,000–340,000	∼65,000	Native transfers only, no
Theoretical peak TPS (mixed, in-memory)	∼125,000–295,000	∼20,000–40,000	Mixed workload 60/20/2
Theoretical peak TPS (persisted)	∼75,000–93,000	∼20,000–40,000	Including RocksDB + N
Sustained TPS (1 shard)	∼10,000–20,000	∼2,000–4,000	Mainnet steady state, inc
Sustained TPS (4 shards)	∼17,000	N/A	ACE architecture suppor
Sustained TPS (8 shards)	∼31,000	N/A	Solana has no native sha
Actual user TPS	—	∼1,000	Solana excluding vote tra
Vote transaction share	0%	∼65%	ACE votes are off-chain
Block limit (MAX_TXS)	80,000	∼2,500–5,000	Architectural Ceiling*
Block size (MAX_BYTES)	32 MiB	1.2 MiB	Accommodates high tx d

Table 16: ACE Chain vs. Solana TPS comparison (peak and sustained)

* The 80,000 TPS figure represents an **Architectural Ceiling** derived from first-principles modelling of the execution pipeline on 32-core server hardware. It is a theoretical upper bound of the architecture's capacity, not a measured benchmark of the current implementation on a geographically distributed mainnet. Actual sustained throughput will be determined by consensus latency, network propagation, and storage I/O in a production environment.

Note: ACE Chain's peak and sustained TPS figures are first-principles derivations based on the cryptographic architecture, modeled on industry-equivalent hardware (32-core AMD EPYC, 128 GB RAM, NVMe SSD, 10 Gbps NIC), with methodology consistent with published benchmarks from Aptos (170K execution-only) and Sui (297K PTB=100). Solana data comes from mainnet measurements (2024–2026). Actual mainnet sustained TPS will be materially lower than model projections, depending on consensus, network propagation, persistent storage, and other engineering factors.

ACE Chain's structural advantage in the TPS dimension derives from four layers:

1. **Reduced cryptographic overhead**: HMAC credential verification (∼1 μs) replaces

Ed25519 signature verification (∼76 μs), freeing CPU resources for transaction execution.

2. $O(1)$ **block verification**: Validator nodes need only verify a single ZK proof rather than per-transaction signatures; verification throughput does not grow with transaction count.

3. **Zero vote transaction overhead**: All TPS capacity serves user transactions, with no protocol-layer consumption.

4. **Linear shard scaling**: HKDF context isolation enables zero-coordination sharding, with TPS growing linearly with shard count.

11.6 In-Runtime DeFi: Near-Zero Fees

In traditional DeFi architectures, every swap, lending, or yield farming operation executes as an independent smart contract call, incurring gas fees, MEV extraction, and cross-contract invocation overhead. Under ACE Chain's architecture, DeFi primitives (such as the Yault ERC-4626 vault) operate within the same runtime's state tree, offering the following cost advantages:

- **No cross-contract invocation overhead**: Yault's `deposit`, `redeem`, and `transfer` operations are state transitions within the runtime, incurring no additional inter-contract communication costs.
- **No MEV extraction**: Transaction ordering is guaranteed by PoH deterministic timestamps, eliminating the arbitrage surface for MEV searchers.
- **Approximately $0.01 per million transactions**: DeFi operations share the same cost structure as ordinary transfers, with no pricing premium for operation complexity.

This property is particularly critical for the Agent economy: when AI agents need to perform asset swaps or redeem liquidity from vaults before executing payments, these DeFi operations incur near-zero fees, preserving the economic viability of micro-transactions.

11.7 Agent Economy Fitness Analysis

Synthesizing the performance and cost characteristics described above, ACE Chain possesses structural advantages in Agent economy scenarios:

Agent Economy Requirement	ACE Chain	Solana	Differential Analysis
Micro-tx economics	∼$0.01/M tx	∼$2.86/M tx	ACE cost ∼300× lower; cent-level tx viable
Finality latency	Target ∼600 ms	∼12 s	Agent negotiation–settlement cycle needs no compensation logic
DeFi integration cost	In-runtime, near-zero	Cross-contract calls, incl. gas	Agent auto-swaps do not erode margins
Policy execution	AESP on-chain policies	No native support	Policy violations frozen on-chain, not audited post-hoc
Identity isolation	HKDF context	Requires multiple wallets	Each

11.8 RWA and Compliant Asset Fitness

Real World Asset (RWA) tokenization imposes a set of requirements on the underlying blockchain that are complementary to those of the Agent economy:

- **Identity compliance**: RWA requires that on-chain identities be traceable to KYC-verified natural persons. ACE-GF's `idcom` commitment can bind KYC status on-chain while protecting personal information from appearing on-chain—only a zero-knowledge proof that "this identity has passed KYC" is exposed on-chain.
- **Inheritance and custody**: Tokenized RWA assets (such as real estate fractions and bond tokens) inherently require inheritance mechanisms. CT-DAP's condition-triggered inheritance paths are directly applicable to RWA scenarios.
- **Low-cost high-frequency settlement**: RWA dividend distributions, interest payments, and similar operations typically involve large volumes of small-value batch transfers. ACE Chain's cost of approximately $0.01 per million transactions makes automated on-chain distribution economically viable.
- **In-runtime DeFi**: RWA tokens within the Yault vault can directly participate in lending and yield generation without cross-protocol bridging. Asset tokenization, custody, yield generation, and compliance close the loop within a single runtime.

12 Security Analysis

12.1 Threat Model

Threat	Defense Mechanism
Key leakage	REV is never persistently stored; SA encrypted with AES-256-GCM-SIV
On-chain identity forgery	`idcom` is a cryptographic commitment; REV cannot be reverse-engineered from on-chain data
Signature forgery	ZK circuit proves credential correctness; computationally unforgeable
Replay attack	Domain binding (chain_id || slot) prevents cross-chain/cross-slot replay
Quantum attack	`idcom` reveals no public key; ML-DSA-44 (Stream 7) can be activated at any time
VA-DAR registration overwrite	First-write binding + monotonically increasing version number
Cross-shard attack	HKDF context isolation guarantees cryptographically disjoint address spaces

Table 18: Threat model and defense mechanisms

12.2 Dual-Algorithm Native Layer: Legacy and PQC in Parallel

ACE Chain does not treat post-quantum cryptography as a future migration—it operates as a **first-class peer** alongside classical algorithms from day one. The Tagged Signature mechanism and identity–authorization separation enable a *dual-algorithm native layer* in which Ed25519 (classical) and ML-DSA-44 (FIPS 204, post-quantum) accounts coexist on the same chain at every protocol level:

- **Consensus layer**: Validator block-production signatures, committee approvals, and finality votes use ML-DSA-44 by default. Ed25519 validators are equally supported; algorithm selection is per-validator at genesis or via on-chain governance.
- **Account layer**: Each on-chain account carries a *tagged public key* (`TaggedPubkey`) recording its algorithm identifier. The attestation verification layer dispatches to the corresponding verifier (Ed25519 or ML-DSA-44) automatically. Ed25519 accounts and ML-DSA-44 accounts coexist without protocol-level branching.
- **Client layer**: The ACE-GF WASM library (`acegf-wallet`) exports both `acegf_sign_message_wasm` (Ed25519, curve = 0) and `ml_dsa_44_sign_wasm` (ML-DSA-44, curve = 2), enabling browser and mobile clients to produce post-quantum signatures natively.

- **Identity continuity**: The identity commitment (idcom) is algorithm-agnostic—it is derived from the ACE-GF identity root, not from any particular signing key. An account can migrate its authorization algorithm (e.g., from Ed25519 to ML-DSA-44 via the OP_SET_AUTH_PUBKEY opcode) without changing its on-chain address.

This design ensures that ACE Chain is production-ready for both algorithm families today, while preserving a zero-cost migration path when quantum threats materialize:

```
Classical: credential = Ed25519-Sign(attest_key_ed25519, msg)
Post-quantum: credential = ML-DSA-44-Sign(attest_key_pqc, msg)
```

Because the on-chain identity is idcom (not a public key), switching the signature algorithm requires only: (1) the client derives the post-quantum key using HKDF Stream 7; (2) an OP_SET_AUTH_PUBKEY transaction updates the account's tagged public key; (3) subsequent transactions use the new algorithm. No account migration, state tree updates, contract redeployment, or address changes are required.

12.3 PQC-Shielded Cross-VM Execution

ACE Chain's n-VM architecture enables an industry-first capability: **users can authorize transactions on any virtual machine—including EVM smart contract calls, SVM program execution, and BVM script operations—using post-quantum signatures (ML-DSA-44), without waiting for those ecosystems to upgrade to post-quantum cryptography themselves.**

12.3.1 Dual-Layer Decoupled Architecture

In traditional blockchains, signature verification and transaction execution are tightly coupled. For example, Ethereum's EVM transaction format (EIP-155/1559) hardcodes Secp256k1 ECDSA signatures (v, r, s) into the RLP structure, making signature algorithm upgrades require coordinated changes across the entire ecosystem.

ACE Chain breaks this constraint by splitting transaction processing into two fully independent layers:

```
+----------------------------------------------------------+
| Authorization Layer |
| +---------------------------------------------------+ |
| | TaggedSignature Verification | |
| | +- Ed25519 (64B) -> Legacy Mode | |
| | +- Secp256k1 (64B) -> Legacy Mode | |
| | +- ML-DSA-44 (2420B) -> PQC Mode | |
| | | |
| | Output: verified sender identity (idcom, 32 bytes) | |
| +---------------------------------------------------+ |
| | |
| +--------------------------------------------------+ |
| | Execution Layer | |
| | +- Native (0x01-04) Native transfers & sys ops | |
| | +- EVM (0x10-12) Solidity contract call/deploy | |
| | +- SVM (0x20-21) Solana program interaction | |
| | +- BVM (0x30-31) Bitcoin script execution | |
| | | |
| | Input: verified idcom -> mapped to VM-native addr | |
| +--------------------------------------------------+ |
+------------------------------------------------------------+
```

After the authorization layer verifies the signature, it passes the authenticated sender identity (`idcom`) to the execution layer. The execution layer's VM engines receive only the verified identity—they are **completely agnostic to the signature algorithm**. Whether a user signs with Ed25519 or ML-DSA-44, the EVM engine sees the same `msg.sender` address.

12.3.2 Dual-Path Compatibility

ACE Chain supports two transaction submission paths simultaneously:

Path	Signature	Security	Wallet Compatibility
Raw Chain	Chain-native (EVM: Secp256k1, SVM: Ed25519)	Legacy	MetaMask, Phantom, etc.
ACE Native	TaggedSignature (incl. ML-DSA-44)	PQC	Yallet, Portal, ACE-GF SDK

Both paths share the same account address and asset balances. Users can interact with DeFi contracts via MetaMask in Legacy Mode for everyday operations, and switch to Yallet in PQC Mode for quantum-secure signing of calls to the same contract—without transferring assets or switching accounts.

12.3.3 Automatic Inheritance Across All VMs

Because PQC capability resides in the authorization layer rather than the execution layer, **all VM engines supported by ACE Chain automatically inherit post-quantum signature protection** without any VM-level modifications:

VM Engine	Target Ecosystem	idcom Address Mapping	PQC
EVM	Ethereum / ERC-20 / DeFi	idcom → 20-byte EVM address	✓
SVM	Solana programs	idcom → 32-byte Solana pubkey	✓
BVM	Bitcoin Script	idcom → 33-byte compressed key	✓
TVM	Tron contracts	idcom → 20-byte Tron address	✓
Native	ACE native operations	idcom used directly	✓

This design also applies to **any future VM engines**. A new VM need only implement the standard `idcom → VM-native address` mapping function to automatically inherit full PQC signature verification—achieving "zero-cost quantum-safe extension." ACE Chain's post-quantum defense capability is fully orthogonal to VM ecosystem expansion: the security layer is built once, and all execution environments benefit permanently.

12.3.4 Security Implications and Competitive Advantage

The practical impact of this architecture is profound:

- **Instant quantum protection**: Users can protect their EVM contract interactions, DeFi operations, and high-value assets deployed on ACE Chain with ML-DSA-44 signatures today, without waiting for any external ecosystem's PQC upgrade.
- **Progressive migration**: Institutional users can migrate critical business contracts from traditional chains to run on ACE Chain, progressively gaining quantum-safe guarantees. Contract code (Solidity/Rust) requires no modification; migration cost is minimal.
- **Cross-VM consistency**: A single ML-DSA-44 key pair can protect a user's operations across all VM engines (EVM, SVM, BVM, TVM) on ACE Chain—achieving true "one key, multiple VMs, quantum-safe."
- **Signature unforgeability upgrade**: Even an attacker with quantum computing capability cannot forge ML-DSA-44 signatures to invoke a user's contracts on ACE Chain—a security guarantee no current EVM-compatible chain can offer.

12.3.5 Comparison with Traditional PQC Upgrade Paths

Important boundary: ACE Chain's PQC protection applies to transactions on ACE Chain itself. PQC-signed EVM transactions execute within ACE Chain's n-VM, *not* on Ethereum mainnet—Ethereum's transaction format hardcodes Secp256k1 ECDSA and cannot recognize PQC signatures. Similarly, SVM/BVM transactions on ACE run in ACE's execution environment, not on Solana/Bitcoin mainnet. Interaction with other public chains still requires cross-chain bridges; the bridge's counterparty side uses that chain's native signature scheme, but assets and operations on the ACE side are always PQC-protected.

This is precisely ACE Chain's core value proposition:

Under the traditional path, each public chain seeking post-quantum security must undergo protocol-level refactoring, consensus-layer upgrades, ecosystem-wide migration, and hard-fork coordination—Ethereum began discussing PQC migration in 2024, with the most optimistic estimates placing completion around 2028 or later.

ACE Chain offers an **immediately available alternative**:

- **Zero-modification contract migration**: Developers can deploy Solidity contracts to ACE Chain's EVM engine as-is—contract logic and ABI interfaces remain identical, but user interactions immediately gain PQC signature protection.
- **Zero execution-layer overhead**: PQC signature verification occurs before transactions enter the VM; the VM's internal execution path is identical to that of classical signatures, adding no gas overhead or execution latency.
- **Superior architectural efficiency**: If traditional chains were to implement PQC natively, they would need to verify 2420-byte ML-DSA-44 signatures at the consensus layer, directly reducing block throughput. ACE decouples PQC verification from VM execution, preventing signature verification overhead from becoming an execution-layer bottleneck.

In short: **for scenarios requiring quantum safety, users need not wait for Ethereum or Solana to complete multi-year PQC upgrades—migrating workloads to ACE Chain provides post-quantum protection today, with a familiar EVM/SVM development experience.**

12.4 HMAC Credential Security Boundary

HMAC-SHA256 credentials employ a security model distinct from that of traditional digital signatures:

- **Symmetric key property**: The `attest_key` is a symmetric HMAC key that is never stored on-chain or transmitted over the network.
- **Phase 1a deferred verification**: Validator nodes do not hold the HMAC key and therefore skip immediate verification for this credential type (handled similarly to raw-chain transactions).
- **Phase 2 ZK verification**: The ZK circuit proves in zero knowledge that "the prover holds `attest_key` and the HMAC computation is correct."
- **Spam transaction defense**: Phase 1a's other checks (payload binding, domain binding, identity registration verification) combined with economic penalty mechanisms provide the first line of defense.

13 Conclusion

Over the past fifteen years, the vast majority of technical innovations in the blockchain industry—faster consensus protocols, higher parallelism, more efficient storage engines—have been engineering optimizations atop the fundamental assumption that “public key equals identity.” ACE Chain chooses to question this assumption itself.

ACE-GF’s identity–authorization separation is not an isolated cryptographic technique; it is a generative principle. From this principle, the performance bottleneck ($O(1)$ verification), the asset management dilemma (the self-custody–inheritance–yield trilemma), usability deficiencies (human reachability), and scalability limitations (zero-coordination sharding) are resolved simultaneously—not through independent solutions, but as natural corollaries of a single design decision.

When identity is no longer equivalent to the private key, self-custodied assets can simultaneously generate yield and be inheritable. When authorization instruments are replaceable, the cost of signature verification can be compressed from $O(n)$ to $O(1)$. When cryptographic identity is independent of algorithm choice, the quantum threat dissolves into a simple parameter switch. When DeFi primitives operate within the runtime itself, fees approach zero, and AI agents’ micro-economic actions become economically viable on-chain.

ACE Chain’s architecture simultaneously addresses two rapidly emerging markets: the **Agent economy** (high-frequency micro-payments, policy boundaries, sub-second finality) and **real-world asset tokenization** (identity compliance, inheritance mechanisms, low-cost batch settlement). This is no coincidence—both demand the flexibility that identity–authorization separation provides.

This is the central thesis of ACE Chain: **Not running faster from the same starting point, but standing on a different one altogether.**

References

[1] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, B.-Y. Yang. *Ed25519: High-speed high-security signatures.* 2012.

[2] E. Ben-Sasson, I. Bentov, Y. Horesh, M. Riabzev. *Scalable, Transparent, and Post-Quantum Secure Computational Integrity.* IACR ePrint 2018/046.

[3] Facebook (Polygon). *Winterfell: A STARK Prover and Verifier for Arbitrary Computations.* https://github.com/facebook/winterfell, 2024.

[4] H. Krawczyk. *Cryptographic Extraction and Key Derivation: The HKDF Scheme.* CRYPTO 2010.

[5] NIST FIPS 204: *Module-Lattice-Based Digital Signature Standard (ML-DSA).* 2024.

[6] J. Boman, S. Enginyer, et al. *EIP-4626: Tokenized Vault Standard.* Ethereum Improvement Proposals, 2022.

[7] J. S. Wang. *ACE-GF: A Generative Framework for Atomic Cryptographic Entities.* arXiv:2511.20505, 2025.

[8] J. S. Wang. *ZK-ACE: Identity-Centric Zero-Knowledge Authorization for Post-Quantum Blockchain Systems.* arXiv:2603.07974, 2026.

[9] J. S. Wang. *AR-ACE: ACE-GF-based Attestation Relay for PQC—Lightweight Mempool Propagation Without On-Path Proofs.* arXiv:2603.07982, 2026.

[10] J. S. Wang. *ACE Runtime—A ZKP-Native Blockchain Runtime with Sub-Second Cryptographic Finality.* arXiv:2603.10242, 2026.

[11] J. S. Wang. *Condition-Triggered Cryptographic Asset Control via Dormant Authorization Paths.* arXiv:2603.07933, 2026.

[12] J. S. Wang. *VA-DAR: A PQC-Ready, Vendor-Agnostic Deterministic Artifact Resolution for Serverless, Enumeration-Resistant Wallet Recovery.* arXiv:2603.02690, 2026.

[13] J. S. Wang. *AESP: A Human-Sovereign Economic Protocol for AI Agents with Privacy-Preserving Settlement.* arXiv:2603.00318, 2026.

[14] J. S. Wang. *n-VM: A Multi-VM Layer-1 Architecture with Shared Identity and Token State.* 2026 (forthcoming).

[15] J. S. Wang. *HFIPay: Privacy-Preserving, Cross-Chain Cryptocurrency Payments to Human-Friendly Identifiers.* 2026 (forthcoming).

Contact: jason@yeah.app