ACE Chain Technical Whitepaper

A Natively Post-Quantum Secure Layer 1 Blockchain Without Performance Trade-offs

(Draft)

ACE Labs

https://www.acechain.io

Contact: contact@acechain.io

April 8, 2026

Abstract

The quantum computing threat to blockchain has shifted from theoretical speculation to an engineering countdown. When Shor's algorithm gains the capability to break elliptic-curve cryptography, every blockchain built on the "public key equals identity" assumption will face a catastrophic scenario—replacing the signature algorithm is equivalent to replacing every user's on-chain identity. Ethereum began discussing post-quantum migration in 2024, with optimistic estimates placing completion no earlier than 2028; meanwhile, harvest-now-decrypt-later attacks mean that data committed on-chain today is already at risk.

ACE Chain integrates post-quantum cryptography (ML-DSA-44, NIST FIPS 204) natively at the protocol layer, and achieves post-quantum security without performance trade-offs through its identity–authorization separation architecture. Its core cryptographic framework, ACE-GF (Atomic Cryptographic Entity Generative Framework), decouples on-chain identity from public keys: 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. ML-DSA-44 operates as a first-class citizen on par with Ed25519 from day one—users can protect their EVM contract calls, SVM program execution, and BVM script operations with post-quantum signatures today, **without waiting for any external ecosystem to complete its own post-quantum upgrade**.

The unique ZK-ACE architecture eliminates the traditional L1 performance tax from three layers: $O(1)$ block verification replaces per-transaction signature verification, zero on-chain vote transactions free all throughput for user transactions, and PQC signature verification completes asynchronously on GPU without occupying the execution critical path. Architectural modeling projects sustained throughput at 4–8× Solana mainnet (in multi-shard configurations). The same identity–authorization separation design decision simultaneously unlocks: sub-second cryptographic hard finality, 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 seven systemic challenges facing the industry today—led by the quantum computing threat—derived from a single fundamental design decision: **identity–authorization separation**.

1.1 Quantum Computing Threat: Not the Future, but the Present

In 2024, NIST officially published the first batch of post-quantum cryptography standards (FIPS 203/204/205), marking the transition of post-quantum migration from academic discussion to engineering implementation. Concurrently, Google's Willow quantum chip demonstrated breakthroughs in exponential error correction capability. The quantum computing threat to blockchain has shifted from theoretical speculation to an engineering countdown.

Harvest-now-decrypt-later (HNDL) attacks give this threat **immediacy**: attackers can record public keys and signatures from on-chain transactions today and reverse-engineer private keys once quantum computers mature. For high-value accounts, this means public keys exposed today are already at risk.

When Shor's algorithm gains the capability to break elliptic-curve cryptography, every blockchain built on the "public key = identity" assumption will face a catastrophic scenario: **replacing the signature algorithm is equivalent to replacing every user's on-chain identity**. Each 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. Ethereum began discussing PQC migration roadmaps in 2024, with optimistic estimates placing completion no earlier than 2028; Solana currently has no public PQC roadmap.

ACE's solution: ACE Chain does not treat post-quantum cryptography as a future migration—**it operates as a first-class citizen at the protocol layer from day one**. 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), with ML-DSA-44 (NIST FIPS 204) post-quantum signatures fully available at the consensus layer, account layer, and client layer. More importantly, ACE Chain's authorization–execution dual-layer decoupled architecture enables an industry-first capability: users can protect all their operations on EVM/SVM/BVM with ML-DSA-44 signatures—without waiting for those ecosystems to upgrade to post-quantum cryptography themselves (PQC-Shielded cross-VM execution, Section 9 § 9.3). For users with existing classical keys, migrating to a post-quantum algorithm requires only a simple parameter switch—changing the algorithm does not affect account identity, achieving zero-migration-cost transition (Section 3 § 3.6, Section 9 § 9.2).

1.2 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 10.2). Simultaneously, assets continue to generate yield within the Yault ERC-4626 smart vault (Section 10.2). All three requirements coexist naturally within a single identity framework.

1.3 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 5).

1.4 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.

1.5 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 6 § 6.1).

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 5, Section 8). 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 10.3, Section 8).

1.8 Design Goals Summary

Industry Pain Point	Design Goal	A Sc lu ti Pa
Quantum computing threat	Native PQC, zero migration cost, cross-VM quantum shielding	id Si na tu Sh n- VI
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 ZI pr
Addressing and recovery deficiencies	Human-readable payment and recovery	VA DA Pa
Cross-chain identity fragmentation	Unified identity with multi-chain native execution	7- str HI VI
Finality efficiency	Sub-second cryptographic hard finality	ZI fi- na ity ce tif ca
Agent economy infrastructure	High-frequency micro-payments, policy boundaries, zero-fee DeFi	AI ru tir 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	Effect
Native PQC + cross-VM quantum shielding	Identity is `idcom`, not a public key; authorization layer decoupled from execution layer	ML-DSA-44 protects all VM operations from day one
$O(1)$ block verification	Credentials verified in ZK circuits; recursive proof covers entire block	Eliminates signature verification bottleneck
Sub-second hard finality	ZK proofs provide cryptographically deterministic finality	Target ∼600 ms (Solana ∼12 s)
Multi-chain native execution	Same `idcom` derives per-chain keys via HKDF streams	Unified EVM/SVM/BVM/TVM scheduling
Zero-coordination sharding	HKDF context isolation; same identity, different shards, cryptographically independent	TPS scales linearly
Serverless recovery	REV deterministically reconstructed from password and identifier	Trilemma broken
Human-readable addressing	DiscoveryID maps human identifiers to on-chain identity	Payment cognitive overhead reduced to email level

Table 3: The corollary system of identity–authorization separation

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:

$$\text{seed}[i] = \text{HKDF-expand}(\text{IKM} = \text{REV},\ \text{info} = \text{domain\_tag}[i],\ \text{length} = 32) \tag{2}$$

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:

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

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 7).

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 ≠ 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).

3.7 Dual-Algorithm Native Layer: Ed25519 and ML-DSA-44 in Parallel

ACE Chain does not treat post-quantum cryptography as a future migration option—Ed25519 (classical) and ML-DSA-44 (FIPS 204, post-quantum) operate as **peer first-class citizens** at the protocol layer from day one. This capability is achieved through a three-layer algorithm abstraction:

Layer 1: Tagged cryptographic primitives. Every public key and signature carries a 1-byte algorithm identifier (`TaggedPubkey`, `TaggedSignature`). The unified wire format (`alg_tag(1 B)` ‖ `key_len(2 B,` uses a 2-byte little-endian length field to accommodate ML-DSA-44 public keys (1,312 B), making the protocol stack transparent to algorithm choice:

Algorithm	Public Key	Signature
Ed25519	32 B	64 B
ML-DSA-44	1,312 B	2,420 B

ML-DSA-44 signatures (2,420 B) are **38**× the size of Ed25519 signatures (64 B). In traditional chain architectures, this signature inflation would directly impact bandwidth and block throughput—if 2,420-byte post-quantum signatures were transmitted and verified per-transaction on-chain, the security benefits of PQC would come at a severe performance cost. This is precisely the core obstacle preventing other public chains from deploying PQC.

ACE Chain fundamentally bypasses the signature inflation problem through its ZK-ACE architecture: credentials are verified within zero-knowledge circuits, and a single recursive STARK proof covers all transactions in the entire block—**verification cost is $O(1)$, independent of both signature size and transaction count**. Post-quantum signatures are not transmitted or stored on-chain; they are consumed within the proof. This not only eliminates PQC's performance penalty but enables ACE Chain's block verification efficiency to **surpass** traditional L1 chains that use only classical signatures (Section 5).

Layer 2: Multi-algorithm account model. Each on-chain account holds a primary authorization key (`auth_pubkey`) and up to 3 additional keys (`auth_keys`), supporting Ed25519 and ML-DSA-44 coexistence within the same account. Users can switch the primary algorithm at runtime via the `OP_SET_AUTH_PUBKEY` opcode, or add a post-quantum key as backup via `OP_ADD_AUTH_KEY`—**without changing the on-chain address or migrating assets**.

Layer 3: Unified verification dispatch. Transaction verification routes through a single `verify_signature()` dispatch function to the corresponding verifier based on the algorithm tag (Ed25519 uses `ed25519-dalek`, ML-DSA-44 uses a pure Rust `fips204` library). Consensus-layer validator nodes also support dual algorithms—validator block-production signatures, committee approvals, and finality votes can use Ed25519 or ML-DSA-44, with algorithm selection configured per-node at genesis or via on-chain governance. On the devnet, new accounts and validators **default to ML-DSA-44**.

This design ensures that Ed25519 accounts and ML-DSA-44 accounts fully coexist at the con-

sensus layer, account layer, and client layer, without protocol-level branching or hard forks. Empirical validation on the devnet confirms this: Ed25519 and ML-DSA-44 benchmark sessions ran consecutively on the same chain without clean restart, with accounts holding keys for both algorithms simultaneously (Section 5).

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.

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. When the recipient has already bound their XID (Extended Identity) in the HFI-Pay registry, payment completes in a single step:

```
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` $\rightarrow$ `Funded` $\rightarrow$ `Claimed` $\rightarrow$ `Withdrawn`, or `Created` $\rightarrow$ `Funded`

$\to$ `Expired` $\to$ `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 $\to$ VA-DAR $\to$ SA2 $\to$ reconstruct REV $\to$ recover full cryptographic identity.

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

5 Consensus and Finality

5.1 Consensus Model

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

- **BFT voting (Tendermint-style)**: A three-phase Propose $\to$ Prevote $\to$ Precommit protocol with $\frac{2}{3}$ stake-weighted quorum achieves soft finality. Timeouts are maximum wait bounds; when $\frac{2}{3}$ votes arrive early, the round advances immediately—enabling sub-slot soft finality on the happy path (approximately 400 ms with a 400 ms slot).
- **PoH (Proof-of-History)**: A serial SHA-256 hash chain provides verifiable temporal ordering without reliance on wall-clock synchronization, serving as the canonical time source for leader scheduling and event ordering.
- **ZK proofs**: Circle STARK (Stwo) proofs provide cryptographic hard finality (post-quantum secure, transparent setup).

5.2 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	ZK proof verification	Cryptographic certainty (mathematically unforgeable)

Table 7: 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). When a builder times out after $K = 3$ slots ($K \times 400\,\text{ms} = 1.2\,\text{s}$), the state transitions to BackupWait (builder slashed); if a backup prover submits a proof within $K' = 3$ additional slots, the state advances to Hard; if all parties time out (total timeout $(K + K') \times 400\,\text{ms} = 2.4\,\text{s}$), the state becomes RolledBack (transactions re-enter the pool).

5.3 $O(1)$ 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 $O(1)$.

$$T_{\text{verify}} \approx 1.1\ \text{ms (Circle STARK verification), independent of transaction count} \tag{5}$$

The ZK circuit (ZK-ACE [8]) asynchronously proves the following compound statement: "For each transaction in this block, there exists a witness $(\text{REV}, \text{salt}, \text{ctx}, \text{nonce})$ such that (C1) the identity commitment recomputed from $(\text{REV}, \text{salt}, \text{domain})$ equals the on-chain `idcom`; (C2) the target binding is consistent with deterministic derivation under (REV, ctx); (C3) the transaction hash is authorized under the derived context; and (C4/C5) anti-replay and domain-separation constraints hold." All constraint evaluations use Poseidon2 (a ZK-friendly, post-quantum-safe algebraic hash), not the external-facing HKDF/HMAC primitives used by the wallet layer.

Proof generation proceeds asynchronously on dedicated prover hardware (design target: approximately 240 ms/slot with GPU parallelism; current devnet uses CPU SIMD), pipelined across slots to run in parallel with block production, never blocking the critical path.

5.4 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.

6 Execution Layer: n-VM and Parallel Scheduling

6.1 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-compatible built-in programs (SystemProgram, SPL Token, PDAs)
0x30–0x3F	BVM	Bitcoin Script + UTXO model
0x40–0x4F	TVM	Tron-compatible contracts

Table 8: 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.

Each VM engine operates in **dual-ingress mode**: (1) as an L1 execution engine within ACE Chain, directly processing ACE-native transactions signed with `TaggedSignature` (including ML-DSA-44); (2) as a **raw-chain ingress gateway**, accepting transactions signed in the corresponding external chain's native format (e.g., EVM EIP-1559, Solana Ed25519, Bitcoin SegWit/Taproot, Tron Protobuf). Because all VM engines share a single unified state tree, a cross-VM atomic swap—such as converting an EVM-originated `TokenA` into an SVM-compatible `TokenB`—executes as a single atomic state transition within one block, with zero bridging fees and no external relayer trust assumptions. Developers can deploy Solidity contracts or Solana programs as-is to ACE Chain, enjoying ACE's post-quantum security and sub-second finality while maintaining full compatibility with native chain toolchains (Hardhat, Anchor, etc.).

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.

Atomic Cross-VM Settlement. Because all VM engines execute against a single unified state tree, cross-chain token operations that traditionally require multi-step bridge protocols become atomic state transitions. Consider a user holding `TokenA` on the EVM engine who wishes to obtain `TokenB` on the SVM engine. On ACE Chain, this proceeds as follows: (1) the EVM engine wraps `TokenA` into `wTokenA` via a deposit to the internal bridge contract; (2) the n-VM dispatcher atomically converts `wTokenA` to `wTokenB` through an in-runtime swap against the unified liquidity pool; (3) the SVM engine unwraps `wTokenB` into the user's SVM-compatible balance. All three steps execute within a single block as one atomic state transition—either all succeed or all revert. This eliminates bridging fees, relayer trust assumptions, and the multi-minute latency inherent in cross-chain message passing. The approach generalizes to any pair of supported VM engines (EVM↔SVM, BVM↔EVM, etc.).

6.2 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.

6.3 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).

7 State Sharding: HKDF Context Isolation

7.1 Sharding Principle

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

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

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

7.2 Shard Routing

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

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

7.3 Scalability Projections

Shards	Independent TPS	Shared TPS	Total	Relative to Solana
1	∼5,000	—	∼5,000	∼1.25×
2	∼7,000	∼3,000	∼10,000	∼2.5×
4	∼14,000	∼3,000	∼17,000	∼4.25×
8	∼28,000	∼3,000	∼31,000	∼7.75×

Table 9: 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 5,000 sustained TPS represents a directional estimate; the peak-to-sustained decay ratio (∼7%) is within the normal industry range.

8 Performance and Cost Analysis

8.1 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 (ZK 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 10: ACE Chain vs. Solana pipeline comparison

8.2 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 TPS approximately 5,000; annual transaction volume approximately 157.7 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 11: 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.

8.3 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)	∼136,000	∼65,000	Native transfers only, no persistent
Theoretical peak TPS (persisted)	∼30,000–75,000	∼20,000–40,000	Including NVMe state writes
Sustained TPS (1 shard)	∼5,000	∼3,000–4,000	Mainnet steady state, including co
Sustained TPS (4 shards)	∼17,000	N/A	ACE architecture supports linear s
Sustained TPS (8 shards)	∼31,000	N/A	Solana has no native sharding
Actual user TPS	—	∼1,000	Solana excluding vote transactions
Vote transaction share	0%	∼65%	ACE votes are off-chain messages

Table 12: ACE Chain vs. Solana TPS comparison (peak and sustained)

Note: ACE Chain's peak and sustained TPS figures are projections based on architectural modeling; Solana data comes from mainnet measurements (2024–2026). ACE's single-shard sustained TPS projection (∼5,000) uses conservative assumptions; the peak-to-sustained decay ratio (∼7%) is within the normal industry range.

8.3.1 Devnet Early Benchmark

To verify the stability of the engineering implementation, we deployed a 3-node BFT devnet on a single MacBook Pro (Apple M3, 12 cores, 36 GB RAM) and conducted sustained stress testing with 1,024 concurrent sender lanes. We ran two independent benchmark sessions on the same chain: one with **ML-DSA-44 (NIST FIPS 204) post-quantum signatures** and one with **Ed25519 (classical) signatures**—both are full end-to-end pipelines including client-side signing, on-chain verification, consensus, and finality. The two runs share identical consensus, execution, and ZK/STARK pipelines; only the signature algorithm differs.

The load generator employs an *adaptive TPS probing* strategy: it progressively increases the submission rate from an initial 300 TPS toward a ceiling of 800 TPS, with automatic back-off when back-pressure is detected. Transactions are distributed round-robin across all three RPC

endpoints. The consensus layer uses an adaptive proposal budget that self-regulates block size based on measured execution time. All consensus-critical messages (proposals, prevotes, precommits, and commit certificates) are delivered via a dedicated libp2p request-response protocol with direct peer-to-peer connections, bypassing gossipsub entirely to eliminate yamux head-of-line blocking under high transaction load.

Metric	**Value**	**Notes**
Sustained TPS	574.5	Average over stable blocks (TPS $\geq$500)
Peak TPS	769	Per-block maximum during sustained operation
Blocks committed	1,895	Zero consensus stalls after genesis
Avg tx per block	554	Non-empty blocks; max 750 (budget cap)
Block interval	$\sim$993 ms	Average; range 459–2,364 ms
Signature verification	parallel	Rayon-based multi-core ML-DSA-44 batch verify

Table 13: Devnet PQC stress test results (single laptop, 3 nodes sharing resources, ML-DSA-44 signatures)

Metric	**Value**	**Notes**
Sustained TPS	581.8	Average over stable blocks (TPS $\geq$500)
Peak TPS	794.5	Per-block maximum during sustained operation
Signature verification	parallel	Rayon-based multi-core Ed25519 batch verify

Table 14: Devnet Ed25519 stress test results (same hardware, same chain, Ed25519 signatures)

Multi-algorithm coexistence: Both benchmark sessions ran on the *same chain* without clean restart. Accounts that had previously been provisioned with ML-DSA-44 keys added Ed25519 keys via `OP_ADD_AUTH_KEY`, and vice versa. The per-algorithm auth key bootstrapping logic correctly routes verification to the appropriate key for each algorithm, confirming that Ed25519 and ML-DSA-44 accounts fully coexist at the consensus, execution, and state layers.

Key parameters for this benchmark run:

Parameter	**Value**	**Purpose**
`BLOCK_INTERVAL_MS`	400 ms	Minimum inter-block pacing
`PROPOSAL_TX_INITIAL_BUDGET`	512	Starting tx budget per proposal
`PROPOSAL_TX_MAX_BUDGET`	750	Adaptive budget ceiling
`PROPOSAL_BUILD_TARGET_MS`	1,500 ms	Target wall-clock for block build
Consensus delivery	P2P direct	All votes & proposals via request-response
Sender lanes	1,024	Concurrent ML-DSA-44 signers

Table 15: Core consensus and networking parameters for the PQC benchmark

Key observation: After genesis startup, every block committed at Tendermint round 0 with

zero consensus stalls across 1,895 blocks (∼30 minutes of sustained load). The direct peer-to-peer consensus delivery protocol eliminates gossipsub contention entirely, maintaining stable consensus liveness even at peak throughput. The sustained 574.5 TPS (ML-DSA-44) and 581.8 TPS (Ed25519) on a resource-constrained single-laptop devnet confirm that the throughput bottleneck lies in consensus and networking, not in signature verification—ACE Chain's architecture absorbs PQC overhead without sacrificing throughput.

Benchmark date: 2026-04-08. Source: tag `20260408_DUAL_ALGO_BENCH`.

Note: The above results reflect *engineering stability*, not production performance limits. The 3 consensus nodes and the load generator share a single laptop's CPU, memory, and I/O resources, severely constraining actual throughput due to hardware contention. For production-environment performance expectations with dedicated server deployments, see Table 12.

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.

8.4 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.

8.5 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

8.6 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.

9 Security Analysis

9.1 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 17: Threat model and defense mechanisms

9.2 Dual-Algorithm Native Layer: Classical 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.

9.3 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.**

9.3.1 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.

9.3.2 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.

9.3.3 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.

9.3.4 Security Implications and Competitive Advantage

- **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.

9.3.5 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.**

9.4 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-ACE circuit proves in zero knowledge that the prover holds a valid REV whose Poseidon2 commitment matches the on-chain `idcom`, and that the transaction is authorized under deterministic derivation and anti-replay constraints (ZK-ACE Constraints C1–C5). The circuit uses Poseidon2—a ZK-friendly, post-quantum-safe algebraic hash—rather than HMAC/HKDF, which are external-facing wallet-layer primitives not suitable for efficient in-circuit evaluation.
- **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.

10 Ecosystem

ACE Chain's identity–authorization separation architecture and post-quantum security capability provide a unique infrastructure for higher-level applications. This section introduces the three core ecosystem projects built on ACE Chain.

10.1 Yallet: Post-Quantum Secure Multi-Chain Wallet

Yallet is a browser extension wallet (Chrome Manifest V3) built on the ACE-GF cryptographic framework, serving as the entry point to the ACE Chain ecosystem. Its core cryptographic operations execute within a Rust-compiled WASM sandbox; private keys never leave the sandbox boundary.

Post-quantum signatures. Yallet natively supports ML-DSA-44 signatures at the client layer—the WASM module exports `ml_dsa_44_sign_wasm` (producing 2,420-byte signatures) and `ml_dsa_44_pubkey_hex` (deriving 1,312-byte public keys), and exposes post-quantum signing capability to connected dApps via the `yallet_signMlDsa44` RPC method. This enables users to initiate post-quantum secure transactions directly from the browser.

Multi-chain support. Yallet provides unified management of assets across 10+ chains including Solana, Ethereum, Polygon, Arbitrum, Base, Optimism, BSC, Avalanche, Fantom, and Cronos, with integrated Jupiter (Solana DEX), 1inch/0x (EVM DEX) aggregators and LI.FI cross-chain bridge, as well as Lido (ETH staking) and Marinade (SOL staking).

VA-DAR decentralized recovery. Yallet implements the complete VA-DAR protocol client: users can recover their full cryptographic identity using only a password and email via the on-chain registry, with backup artifacts (SA2) encrypted and stored on Arweave, requiring no centralized server.

ZK-ACE zero-knowledge authorization. Yallet integrates STARK proof generation capability, supporting both nonce and nullifier replay protection modes, proving wallet ownership and authorizing transactions without exposing private keys.

Encrypted NFT storage (RWA). Yallet treats personal data as on-chain assets: users' photos, documents, notes, and contacts are encrypted via ECIES and uploaded to Arweave for permanent storage, while compressed NFTs (cNFTs, based on Metaplex Bubblegum) are minted on Solana as on-chain indices. Encryption keys are derived from the user's xidentity (Ed25519 public key); only the user holding the corresponding private key can decrypt—even though the data on Arweave is fully publicly readable, its contents are cryptographically indistinguishable from random bytes. cNFT minting costs approximately 5,000 lamports (under $0.001), and Arweave storage is one-time payment with permanent retention, achieving a combination of **absolute privacy and unlimited storage**.

Invoice system. Yallet includes built-in B2B invoice management: supporting creation of structured invoices with line items, tax rates, and discounts, exportable as PDF (with encrypted payment links) and CSV formats. Invoices support both fiat (USD, EUR, etc.) and

cryptocurrency (USDC, SOL, ETH, etc.) denomination. When sending an invoice, the system simultaneously mints an encrypted cNFT for both sender and recipient, enabling end-to-end encrypted invoice delivery.

Verifiable Credentials (VC). Yallet implements the W3C Verifiable Credentials Data Model 2.0 standard using `eddsa-jcs-2022` data integrity proofs. The DID method is `did:yallet` (derived directly from the wallet's xidentity), with additional support for resolving `did:key` and `did:web` (institutional issuers). Selective disclosure is achieved through per-claim salt commitments: issuers generate independent salts and SHA-256 commitments for each claim; holders can choose which claims and their salts to disclose; verifiers can only verify the correctness of disclosed claims, while undisclosed claims exist only as hashes. This enables users to prove specific attributes (e.g., "age $\geq$ 18" or "KYC passed") to third parties without exposing their full identity.

10.2 Yault: Self-Custody Asset Management Platform

Yault (`yault.xyz`) is a self-custody asset management platform that simultaneously addresses yield generation and inheritance for crypto-assets. Industry estimates suggest over $100 billion in crypto-assets have been permanently lost due to holder deaths—Yault breaks the self-custody–inheritance–yield trilemma within a single framework through the fusion of CT-DAP (Condition-Triggered Dormant Authorization Paths) and ERC-4626 smart vaults.

ERC-4626 smart vault. After users deposit assets into a vault based on the ERC-4626 standard, they automatically connect to Aave V3 lending pools for yield generation. Yield is distributed at a 75/25 ratio (75% auto-compounded to the user, 25% to the platform). Core operations include `deposit`, `redeem`, and `transfer`.

CT-DAP condition-triggered inheritance. CT-DAP creates a dormant authorization path on top of identity–authorization separation. Its core is a three-factor credential mechanism: on-chain release proof (Seal Authority), beneficiary passphrase (8-word phrase), and management factor (256-bit key)—no single party can independently access the assets. Chainlink CRE executes four independent verifications in parallel (drand cryptographic timestamp, vault balance, KYC/AML compliance screening, price oracle) before a release proof can be submitted. Assets continue to generate yield in the vault throughout the dormancy period.

Deep Chainlink integration. Yault integrates five Chainlink services: CRE (oracle proof pipeline), Data Feeds (real-time portfolio valuation), Automation (automatic yield harvesting), CCIP (cross-chain proof relay), and Functions (off-chain risk analysis).

Agent Spending Policies. Yault supports setting budgets and operational boundaries for AI agents (daily/monthly limits, address whitelists, time windows), enforced on-chain.

10.3 AESP: Agent Economic Sovereignty Protocol

AESP (Agent Economic Sovereignty Protocol) is an Agent economy protocol SDK built on ACE Chain, enabling AI agents to autonomously execute economic actions under human sovereignty

constraints.

Policy engine. Every economic action initiated by an agent must pass 8 deterministic checks before execution: per-transaction limit, rolling budget (daily/weekly/monthly), address whitelist, chain whitelist, time window, first-payment review, minimum balance, and budget audit. Any check failure automatically escalates to the human review queue.

Negotiation and commitments. AESP provides a state-machine-driven inter-agent negotiation flow (offer/counter-offer/accept/reject) and EIP-712 structured commitments, supporting on-chain escrow settlement.

Synergy with ACE Chain. Sub-second hard finality enables an agent's negotiation–settlement cycle to complete within a single slot (400 ms); HKDF context isolation generates a cryptographically independent address for each agent transaction; near-zero fees (approximately $0.01 per million transactions) make cent-level micro-transactions economically viable.

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 external agents.

11 Conclusion

The quantum computing threat to blockchain is not a problem that can be deferred to tomorrow. Harvest-now-decrypt-later attacks mean that public keys exposed on-chain today are already at risk, while the post-quantum migration roadmaps of existing mainstream public chains remain at the discussion stage. ACE Chain provides an immediately available answer: **natively supporting post-quantum cryptography at the protocol layer, providing users with quantum security guarantees from day one.**

This capability is not achieved through patching, but stems from a fundamental architectural decision—identity–authorization separation. When on-chain identity (`idcom`) is no longer bound to any specific signature algorithm, post-quantum migration transforms from a catastrophic upgrade involving chain-wide state changes into a simple parameter switch. More importantly, the decoupling of the authorization layer from the execution layer enables ML-DSA-44 signatures to protect operations across all VM engines—users need not wait for Ethereum or Solana to complete multi-year PQC upgrades; migrating workloads to ACE Chain provides post-quantum protection today.

The same identity–authorization separation principle simultaneously unlocks: $O(1)$ block verification, sub-second cryptographic hard finality, breaking the self-custody–inheritance–yield trilemma, human-readable payment addressing and wallet recovery, zero-coordination state sharding, and near-zero-fee settlement infrastructure for the Agent economy and RWA tokenization. These are not independent solutions, but natural corollaries of a single design decision.

This is the central thesis of ACE Chain: **Not running faster from the same starting point, but standing on a starting point that is cryptographically prepared for the post-quantum era.**

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: contact@acechain.io