VOLUME I · 旗舰协议与形式化技术白皮书

# MYRILUM 灵核OS公链（Linghe Chain）：面向自主智能体商业与全链合规清算的亚秒级并行主权公链协议与形式化技术白皮书（v2.0）

亚秒级并行共识、双椭圆曲线密码学信任根、内生监管原语与万链双向清算的形式化协议规范

MYRILUM 灵核公链联合院士编纂委员会（分布式系统 · 应用密码学 · 央行货币工程 · 金融监管科技 · 算法博弈论）




| 单槽确定性终局 | 二维并发执行通道 |
|---|---|
| 250 ms | 32 Lanes (0x76) |
| Commonware Simplex BFT · RPO = 0 | Tempo Reth EVM · 3.0B Block Gas |
| **零知识配对跨链核验** | **硅基硬件飞地信任根** |
| 815,381 gas | 0x0100 Low-s P-256 |
| BN254 Groth16 Pairing · 0.027% Slot | Osaka EIP-7951 Silicon Root of Trust |

协议与软件系统研发主体

**MYRILUM LIMITED**

中华人民共和国香港特别行政区《公司条例》（第 622 章）注册成立之法人团体（Body Corporate）

CR / UBI: 80912065 · D-U-N-S: 374308629
BR Cert: 80912065-000-07-26-0
Release: v2.0 Canonical Edition (2026)

REGISTERED OFFICE: UNIT 1603, 16/F THE L. PLAZA, 367-375 QUEEN'S RD CENTRAL, SHEUNG WAN, HONG KONG

# 协议研发主体登记要素、香港法律适用边界与链上存证指纹

VOL. 01 专著卷册定位与文献分级

VOL. I · FORMAL PROTOCOL & CRYPTOGRAPHIC SPECIFICATION

MYR-WP-TECH-2026-V2.0-ZH · Canonical Release v2.0 (2026)

CHAIN 88888 链上密码学存证与安全守卫

LingheSovereignSecurityGuardV5.latestWhitepaperAnchor()

Simplex BFT 250 ms · Verify Endpoint: `/v1/whitepapers/verify`

| 研发主体登记与密码学核验维度 | 官方核实唯一权威真值（**SSOT**） |
|---|---|
| 协议与软件系统研发主体 | `MYRILUM LIMITED` (香港法例第 622 章《公司条例》私人股份有限公司) |
| 公司编号 / **UBI** 与邓白氏编码 | CR/UBI: `80912065` · D-U-N-S®: `374308629` |
| 商业登记证号码（第 **310** 章） | `80912065-000-07-26-0` |
| 香港法定注册办事处 | UNIT 1603, 16/F THE L. PLAZA, 367-375 QUEEN'S RD CENTRAL, SHEUNG WAN, HONG KONG |
| 国际标准文献编号 | `MYR-WP-TECH-2026-V2.0-ZH` · PUBLIC PROTOCOL TECHNICAL WHITEPAPER · SOVEREIGN CHINESE EDITION (v2.0) |
| 公链标识与不可变创世块摘要 | ChainID `88888` ( `0x15b38` ) · Genesis SHA-256: `96208e9900b218935143be3024d2602566a86d61caa906d42399f35b83afd991` |
| **LHKD** 合成基准默克尔状态根 | `0xa3e5fd1ee51db26128c49f848bc4eb6d71443fbfee51a40b9159832a21beae6e` (101.20% · 78,936,000.00 Synthetic Benchmark Vector) |
| **LUSD** 合成基准默克尔状态根 | `0x5c3f82332fbffc0be49434232a9a97e2bf2c797423aa3fde4afd19b2c264f410` (100.50% · 10,050,000.00 Synthetic Benchmark Vector) |
| 本卷 **Markdown** 母本 **SHA-256** | `5bf2bec72cdda421c44854027d6776ff9596d11d72762c93caa5024828a32818` |

**香港法律适用边界与纯底层技术规范声明（第 32、571、656、615、155/581 及 486 章）**

1. 纯底层技术规范与非要约边界：本专著系由 MYRILUM LIMITED 仅以「协议与软件系统研发主体」身份编纂之分布式系统与密码学工程技术规范，不构成香港《公司（清盘及杂项条文）条例》（第 32 章）招股章程、《证券及期货条例》（第 571 章）证券或集体投资计划要约、《稳定币条例》（第 656 章）第 8 及第 9 条下指明稳定币之发行、要约或积极推广，亦不构成第 615 章虚拟资产服务。LHKD、LUSD 与 LHC 均为协议内生标准账本原语与非流通确定性基准测试状态。

2. 零利息非存款、终端非托管与隐私保护不变量：底层状态机在字节码层恒定锁定持有端零利息（r_holder(t) === 0，非第 155/581 章银行存款）；MYRILUM LingheOS Wallet 为 100% 客户端本地自托管零第三方运行时依赖软件；链上账本严格遵循《个人资料（隐私）条例》（第 486 章）实现零自然人隐私数据上链。

DOCUMENT NAVIGATION · 全卷结构化目录索引

# 全卷章节目录与形式化结构索引

瑞士国际主义网格学术目录 —— 支持在电子阅读器中点击任意章节目次跳转至对应形式化规范与数学证明章节。


00 摘要（**Executive & Scientific Abstract**） §00

00·1 全局数学符号与密码学原语公理表（**Global Mathematical Notation & Primitives**） §00·1

01 第一性原理与系统架构动机 §01

› 1.1 自主主权清算不可能三角的形式化建模（The Autonomous Sovereign Settlement Trilemma） §1.1

› 1.2 一维串行 EVM、12 秒概率槽位与应用层合规包装器的定量批判 §1.2

› 1.3 七层主权架构与 `9+6+2` 合约及预编译全景映射 §1.3

02 世界状态空间与全局状态转移总算子 §02

› 2.1 六维正交世界状态元组与不可变创世零块承诺 §2.1

› 2.2 全局区块状态转移总算子与内核预编译分派拓扑 §2.2

03 **Commonware Simplex BFT 250ms** 流水线共识与跨法域容错证明 §03

› 3.1 四法域主权验证委员会拓扑与部分同步对抗网络模型 §3.1

› 3.2 三阶段流水线槽位动力学与虚拟空块无停顿跳槽机制 §3.2

› 3.3 可问责安全性（Accountable Safety）与有界亚秒级活性形式化证明 §3.3

› 3.4 复合对抗边界分析：跨洲海缆中断叠加提案双签与抢跑操纵防御 §3.4

04 **Tempo Reth** 并行执行引擎与 **0x76** 二维随机数无锁代数 §04

› 4.1 模块化 Rust 零拷贝执行管线与二维随机数状态流形 §4.1

› 4.2 `0x76` 原生交易信封、原子多调用批处理与 Passkey 密钥链授权 §4.2

› 4.3 多通道阿贝尔交换性证明与高并发突发结算实证 §4.3

› 4.4 稳定币代付五阶段价值守恒与并发对抗边界分析 §4.4

05 内核级金融与监管状态机原语（**TIP-20 · TIP-403 · 原生限价订单簿 DEX**） §05

› 5.1 创世内生 `TIP-20` 双轨账本原语体系与 `TIP-403` 内核策略耦合 §5.1

› 5.2 内核原生中央限价订单簿 `Enshrined StablecoinDEX`（`0xdec0000000000000000000000000000000000000`）撮合代数 §5.2

› 5.3 香港金管局联系汇率走廊（`7.8000`）算术与内核不可绕过性定理 §5.3

06 双椭圆曲线密码学根基：**BN254 Groth16** 零知识桥与 **EIP-7951** 规范化 **P-256** 硬件信任根 §06

› 6.1 `BN254` 扩域塔、`LingheZkLightClientBridge` 与 `815,381 gas` 配对核验推导 §6.1

› 6.2 大阪 `EIP-7951`（`0x0100`）硬件飞地 `PCR0` 证明与 `LingheSovereignSecurityGuardV5` 低阶规范化 §6.2

› 6.3 双椭圆曲线密码学对抗证伪与安全边界矩阵 §6.3

07 自主智能体主权商业协议栈：**ERC-8004** 注册表、**x402** 微支付与可验证推理博弈论 §07

› 7.1 `ERC-8004` 创世合约簇与 `LingheSessionKeyManager` 形式化规范 §7.1

› 7.2 `HTTP 402`（`x402 v1`）挑战—清结算—硬件证明状态机 §7.2

› 7.3 可验证推理与反女巫刷分之博弈论均衡分析 §7.3

08 双向万链意图清算与异构托管状态机（**ERC-7683 EVM** 与 **Solana BPF**） §08

› 8.1 六链双向意图架构、`LingheOmnichainSettler` 与 `Permit2` 协同规约 §8.1

› 8.2 异构托管状态机规范：EVM `LingheOriginEscrow7683` 与 Solana `linghe_solana_escrow_7683` §8.2

› 8.3 `v5.0` 全栈对抗审计加固之四重纵深防御机制 §8.3

› 8.4 跨链价值守恒与本金绝对可退性形式化证明 §8.4

09 跨四法域主权门限治理、强制时间锁守卫与二进制储备证明（**Merkle PoR**） §09

› 9.1 跨四法域 `3-of-4` 拜占庭门限治理与硬件密钥信封隔离 §9.1

› 9.2 后创世强制时间锁守卫模块 `LingheSovereignSecurityGuardV5` 与双阶段治理方程 §9.2

› 9.3 香港《稳定币条例》（Cap. 656）参考偿付状态模型与四叶子二叉 Merkle 证明 §9.3

10 零依赖终端供应链安全、双轨反洗钱引擎与金融级 **Fail-Closed** 可观测性 §10

› 10.1 零第三方运行时依赖终端隔离、169 链密码学栈与硬件安全隔区门禁 §10.1

› 10.2 双轨 `FATF Recommendation 16` 旅行规则、24 小时滑动窗口反拆分引擎与原子预写日志 §10.2

› 10.3 边缘网关攻击面收敛、恒定时间管理鉴权与金融级 `Fail-Closed` 遥测公理 §10.3

11 链上实证性能基准与六代形式化证伪矩阵 §11
› 11.1 链上实证延迟、并行微支付吞吐与跨网络量化对比 §11.1
› 11.2 六代全栈对抗演进与 36/36 可证伪不变量核验矩阵（v1.0–v5.1） §11.2
12 结论 §12
AP·A 创世系统合约、内核预编译与后创世安全模块规范总表 §A
AP·B 核心协议数据包与密码学域分离字节级编码规范 §B
› B.1 LINGHE_CAP656_HQLA_V1 储备叶子紧凑原像布局（abi.encodePacked -> keccak256） §B.1
› B.2 LINGHE_HW_ATTESTATION_V3 硬件可信飞地证明摘要原像（定长 168 Bytes -> SHA-256） §B.2
› B.3 LINGHE_PROTECTED_CLIENT_ID_V5 防抢跑客户端订单标识原像（定长 133 Bytes -> keccak256） §B.3
› B.4 0x0100 Osaka EIP-7951 P256VERIFY 预编译调用帧与低阶标量约束（定长 160 Bytes 输入） §B.4
› B.5 Solana Ed25519SigVerify 指令内省头部与 213 Bytes 跨链清偿摘要布局 §B.5
AP·C 国际标准、监管法例与学术参考文献 §C

MYR-WP-TECH-2026-V2.0-ZH · PUBLIC PROTOCOL TECHNICAL WHITEPAPER · SOVEREIGN CHINESE EDITION (v2.0)

# MYRILUM 灵核OS公链（Linghe Chain）：面向自主智能体商业与全链合规清算的亚秒级并行主权公链协议与形式化技术白皮书（v2.0）

协议主权标识：`MYRILUM Linghe Chain` · `ChainID = 88888`（`0x15b38`）· 白皮书版本：`v2.0`（旗舰形式化黄皮书规范）
协议与软件系统研发主体：`MYRILUM LIMITED`（迈瑞伦有限公司 · 香港公司注册编号 / UBI：`80912065` · 商业登记证号：`80912065-000-07-26-0` · 香港法定注册办事处：`UNIT 1603, 16/F THE L. PLAZA, 367-375 QUEEN'S RD CENTRAL, SHEUNG WAN, HONG KONG`）
共识与并行执行内核：`Tempo`（`Paradigm Reth` 模块化 Rust 并行 EVM + `Commonware Simplex BFT`，`250ms` 单槽确定性终局，`3,000,000,000` 区块 Gas 上限）
不可变创世零块承诺（**Genesis Block-0 SHA-256**）：`96208e9900b218935143be3024d2602566a86d61caa906d42399f35b83afd991`

**法定护栏 0.1 — 香港法律边界与纯技术规范声明**

1. 协议与软件系统研发主体定位：本技术白皮书系关于分布式拜占庭共识状态机、二维并行虚拟机执行引擎、双椭圆曲线密码学验证原语及客户端自托管密钥架构之纯粹计算机科学与系统工程规范文献，由依中华人民共和国香港特别行政区《公司条例》（香港法例第 622 章，`Companies Ordinance, Cap. 622`）合法注册成立之私人股份有限公司 `MYRILUM LIMITED`（迈瑞伦有限公司，`BODY CORPORATE`）作为唯一协议与软件系统研发主体（`Protocol & Software R&D Author`）独立研发与编撰（香港公司注册编号 / 唯一商业识别码 CR / UBI：`80912065`，成立日期：`27 July 2026`；《商业登记条例》香港法例第 310 章商业登记证号码：`80912065-000-07-26-0`；香港法定注册办事处：`UNIT 1603, 16/F THE L. PLAZA, 367-375 QUEEN'S RD CENTRAL, SHEUNG WAN, HONG KONG`）。`MYRILUM LIMITED` 严格限定于底层分布式协议与 100% 客户端自托管软件（`MYRILUM LingheOS Wallet`）之纯技术工程研发，绝不接受、持有或托管任何公众法币储备资金，亦从不触碰或控制任何客户端私钥与助记词。
2. 香港法例非要约与非受规管金融活动声明：本文件纯粹作为密码学协议形式化建模与确定性工程基准说明，在任何司法管辖区均不构成《公司（清盘及杂项条文）条例》（香港法例第 32 章，`Cap. 32`）所指之招股章程，不构成《证券及期货条例》（香港法例第 571 章，`Cap. 571`）项下之证券、期货合约、结构性产品或集体投资计划（CIS）认购要约或投资顾问意见，不构成《稳定币条例》（香港法例第 656 章，`Cap. 656`）第 8 条及第 9 条项下之受规管稳定币活动、指明稳定币要约或向公众积极推广，亦不构成《打击洗钱及恐怖分子资金筹集条例》（香港法例第 615 章，`Cap. 615`）项下之受规管虚拟资产服务。
3. 协议内生标准账本原语、确定性基准状态与零个人隐私（`Cap. 486`）准则：创世状态树内生固化之 `LHKD`（`0x20C0000000000000000000000000000000000002`）、`LUSD`（`0x20C0000000000000000000000000000000000001`）与 `LHC`（`0x20C0000000000000000000000000000000000000`）均严格界定为协议内生标准账本原语与非流通确定性基准测试状态（`Enshrined Protocol Ledger Primitives & Non-Circulating Deterministic Benchmark States`），专门用于在状态机字节码层面形式化建模并验证香港 `Cap. 656` 附表 2 全额偿付覆盖率不变量、`Cap. 615` 与 `FATF Recommendation 16` 密码学信封承诺、欧盟 `Regulation (EU) 2023/1114 (MiCA)` 及巴塞尔委员会 `SCO60` 风险隔离数学性质；其中 `LHKD` 与 `LUSD` 在状态机层面强制锁定零生息率不变量（$r_{\text{holder}} \equiv 0$），不含任何利息计提、分红派息、储备收益分配或净值重定基逻辑。全栈代码、链上存证及本文献严格执行香港《个人资料（私隐）条例》（香港法例第 486 章，`Cap. 486`）数据最小化准则，强制维持零自然人个人身份信息（Zero Individual PII）。

第 00 节 · SECTION 00

## 摘要（Executive & Scientific Abstract）

全球金融清算与分布式计算体系正处于两股刚性演进力量的正面交汇点：一方面，互联网微观经济主体正从人类低频手工交互跃迁为全天候自治协作的人工智能体集群（Autonomous AI Agent Swarms），其高频按次调用模型推理、机密算力与实时金融数据的机器原生商业（`HTTP 402 Payment Required`）要求底层账本同时具备亚秒级确定性终局、同金库无队头阻塞的并发微支付吞吐、免波动性母币预持的原生稳定币计费以及硅基可信执行环境（TEE）算力真伪核验；另一方面，以香港《稳定币条例》（`Cap. 656`）及欧盟 `MiCA` 为代表的主权金融立法，强制要求链上法币结算媒介在任何时刻维持 $\geq 100\%$ 破产隔离高流动性优质资产（HQLA）储备、法定汇率走廊约束、不可绕过的制裁实体拦截以及零明文隐私泄露的跨境旅行规则（`FATF Recommendation 16`）合规存证。传统代理行电汇与卡组织批处理网络受困于 `T+1` 至 `T+3` 交收时滞及高于微支付面值数倍的固定清算费率，而第一代顺序执行公链则受制于一维串行 Nonce 在高并发下的排队论发散、12 秒概率性槽位重组敞口、波动性 Gas 代币摩擦以及应用层外挂黑名单在智能合约组合调用下的穿透失效，二者均无法在单一状态机内闭合上述正交约束。

本文提出并形式化证明 **MYRILUM** 灵核**OS**公链（**Linghe Chain**，`ChainID = 88888` / `0x15b38`）——一种面向自主智能体商业与全链合规清算的亚秒级并行主权第一层（Layer-1）确定性状态机。在共识与并行执行层，协议构建于横跨香港（`HK-HKMA`）、新加坡（`SG-MAS`）、东京（`JP-FSA`）与苏黎世（`CH-FINMA`）四大监管法域的 `Commonware Simplex BFT` 三阶段流水线拜占庭共识之上，达成单槽 $\Delta_{\text{slot}} = 250$ ms 零分叉确定性终局；执行引擎基于深度定制的 `Paradigm Reth` Rust 模块化内核，提供每区块 $3,000,000,000$ Gas（即 $1.2 \times 10^{10}$ Gas/s）计算带宽，并通过 `0x76` 二维并行随机数流形 $(k, \nu) \in \mathbb{Z}_{2^{192}} \times \mathbb{Z}_{2^{64}}$（预编译地址 `0x4e4f4e4345000000000000000000000000000000`）与 `TIP Fee Manager` 原生多账本原语费用清算器（`0xfeec000000000000000000000000000000000000`），实测实现 32 通道并发微支付 `20/20` 笔无锁批量结算耗时 `~1,004 ms` 且零 Nonce 碰撞。在协议内生账本原语与内核合规状态机层，创世零块（不可变摘要 `96208e9900b218935143be3024d2602566a86d61caa906d42399f35b83afd991`）将共识质押计量账本原语 `LHC` 与定值清算参考账本原语 `LUSD`（非流通确定性基准负债状态 $10,000,000.00，协议参考储备模型状态 $10,050,000.00，基准偿付覆盖率 `100.50%`）及 `LHKD`（非流通确定性基准负债状态 HK$78,000,000.00，协议参考储备模型状态 HK$78,936,000.00，基准偿付覆盖率 `101.20%`）实施严格的状态空间正交分离，在 Rust 执行内核层内生固化 `TIP-403` 亚微秒级策略拦截（`0x403c0000000000000000000000000000000000`）与香港金管局 `[7.7500, 7.8500] HKD/USD` 联系汇率字节码走廊。

在密码学证明、自治智能体经济与全链互操作维度，灵核OS公链于创世状态树内生集成 `ERC-8004` 主权身份、支付驱动信誉及 `80% / 20%`（技能执行者 / 底层模型版权方）零粉尘原子版税分润托管合约，并通过 `Osaka EIP-7951` 预编译（`0x0000000000000000000000000000000000000100`）叠加严格低阶标量约束 $0 < s \leq \iota n_{\text{P256}}/2 \rfloor$ 原生核验 NVIDIA H100 与 AWS Nitro 硬件飞地证明。跨链层联合 `ERC-7683` 秒级意图清算海关（`0x8900000000000000000000000000000000000001`）、覆盖六大异构源链（Ethereum、OP Mainnet、BNB Smart Chain、Base、Arbitrum One 与 Solana）的防抢跑双向退本托管金库以及 `BN254 Groth16` 零知识轻客户端桥（`0x8900000000000000000000000000000000000002`，单次四配对验证消耗 `815,381 gas`，仅占单块容量 `0.027%`），实测达成异构源链免 Gas 跨链 `x402` 履约端到端耗时 `~936 ms`。配合后创世安全守卫模块 `LingheSovereignSecurityGuardV5`（强制执行 `3-of-4` 主权时间锁 $\text{eta} = t_{\text{queue}} + \Delta_{\text{min}}$ 与 `LINGHE_CAP656_HQLA_V1` 实时链上 Merkle 包含性复算）及零第三方运行时依赖的自托管终端 `MYRILUM LingheOS Wallet`（支持 `169` 条异构公链与 `10` 个零信任 MCP 签名工具），本协议确立了主权级智能体金融基础设施的完整形式化闭环。

形式化公理符号表 · NOTATION

## 全局数学符号与密码学原语公理表（Global Mathematical Notation & Primitives）

为确保全篇公理推导、状态转换方程与安全证明具备严格的一致性与可证伪性，表 0.1 统一界定全栈形式化符号系统及其对应的链上物理真源常量。

| 符号 / 算子 | 数学定义域与代数结构 | 物理语义、协议常量与链上预编译 / 合约锚定 |
|---|---|---|
| $\sigma \in \mathbf{\Sigma}$ | $\sigma = (\sigma_{\text{acc}}, \sigma_{\text{tip20}}, \sigma_{\text{tip403}}, \sigma_{\text{nonce2d}}, \sigma_{\text{dex}}, \sigma_{\text{sys}})$ | 灵核OS公链六维正交世界状态元组，由并行稀疏 Merkle 前缀树承诺为 32 字节状态根 $\mathcal{R}(\sigma) \in \mathbb{B}^{32}$ |
| $\Upsilon_{\text{tx}}(\sigma, T)$ | $\mathbf{\Sigma} \times \mathcal{T}_{0\text{x}76} \to \mathbf{\Sigma}$ | 单笔交易五阶段原子状态转移算子（前置锁费、二维 Nonce 递增、多调用执行、余量退费、验证者清分） |
| $\Upsilon(\sigma, B_h)$ | $\mathbf{\Sigma} \times \mathcal{B}_h \to \mathbf{\Sigma}$ | 高度 $h$ 区块级复合状态转移总算子，$\Upsilon(\sigma, B_h) = \Omega_h(\prod_{i=1}^{m} \Upsilon_{\text{tx}}(\cdot, T_i)(\sigma))$ |
| $\mathcal{H}_{\text{gen}}$, ChainID | $\mathcal{H}_{\text{gen}} \in \mathbb{B}^{32}$, ChainID = 88888（`0x15b38`） | 不可变创世零块 SHA-256 摘要 `96208e9900b218935143be3024d2602566a86d61caa906d42399f35b83afd991` |
| $(n, f, q, \Delta_{\text{slot}})$ | $n = 3f + 1 = 4,\ f = 1,\ q = 2f + 1 = 3$ | `Simplex BFT` 四法域验证委员会规模、最大拜占庭容错数、法定仲裁阈值及单槽终局周期 $\Delta_{\text{slot}} = 250$ ms |
| GST, $\Delta_{\text{net}}$ | $\text{GST} \in \mathbb{R}_{\geq 0}$, $\Delta_{\text{net}} \in \mathbb{R}_{>0}$ | Dwork-Lynch-Stockmeyer 部分同步模型未知全局稳定时间及同步期诚实节点间单向消息传播上界 |
| $G_{\max}$, $\Theta_{\text{gas}}$ | $G_{\max} = 3 \times 10^9$ Gas, $\Theta_{\text{gas}} = 1.2 \times 10^{10}$ Gas/s | 单区块计算资源硬上限（`3,000,000,000`）与每秒持续理论计算吞吐带宽 |
| $(k, \nu)$ | $\{0, \ldots, 2^{192} - 1\} \times \{0, \ldots, 2^{64} - 1\}$ | `0x76` 交易二维并行 Nonce 坐标（通道键 $k$ 与通道内单调序号 $\nu$），由 `0x4e4f4e4345000000000000000000000000000000` 管理 |
| $T_{0\text{x}76}$, $H_{\text{feePayer}}$ | EIP-2718 类型前缀 `0x76` 与代付签名域 `0x78` | 原生批量调用、二维 Nonce、有效时间窗与稳定币 Gas 代付信封，联动 `0xfeec000000000000000000000000000000000000` |
| $\mathcal{T}_{\text{ens}}$ | $\{\text{LHC}, \text{LUSD}, \text{LHKD}\}$（精度均为 $10^6$） | 创世原生 `TIP-20` 协议内生标准账本原语与非流通确定性基准状态集：`0x20C000000000000000000000000000000000000` 至 `0x20C000000000000000000000000000000000002` |
| $\Phi_{403}(\psi, a)$ | $\mathbb{Z}_{2^{64}} \times \mathcal{A} \to \{0, 1\}$ | 内核级合规策略鉴权谓词（`LHKD` 刚性绑定 `Policy #2`），锚定预编译 `0x403c0000000000000000000000000000000000` |
| $q_{\text{BN}}$, $r_{\text{BN}}$ | 254-bit Barreto-Naehrig 素数基域与标量域阶 | $q_{\text{BN}}$ = 21888242871839275222246405745257275088696311157297823662689037894645226208583；$r_{\text{BN}}$ = 21888242871839275222246405745257275088548364400416034343698204186575808495617 |
| $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ | $e: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T \subset \mathbb{F}_{q_{\text{BN}}}^{\times 12}$ | `BN254`（`alt_bn128`）最优 Ate 双线性配对群系，锚定预编译 `0x0000000000000000000000000000000000000006` / `0x0000000000000000000000000000000000000007` / `0x0000000000000000000000000000000000000008` |

| 符号 / 算子 | 数学定义域与代数结构 | 物理语义、协议常量与链上预编译 / 合约锚定 |
|---|---|---|
| $n_{\mathrm{P256}}$, $\lfloor n_{\mathrm{P256}}/2 \rfloor$ | NIST `secp256r1` 素数子群阶与规范半阶上界 | $n_{\mathrm{P256}}$ = `0xFFFFFFFF00000000FFFFFFFFFFFFFFFFBCE6FAADA7179E84F3B9CAC2FC632551`；半阶 $\lfloor n_{\mathrm{P256}}/2 \rfloor$ = `0x7FFFFFFF800000007FFFFFFFFFFFFFFFDE737D56D38BCF4279DCE5617E3192A8`，调用 `0x0000000000000000000000000000000000000000000000000000000000000100` |
| $\mathbb{G}_{\mathrm{Ed25519}}$, $\mathbb{G}_{\mathrm{k1}}$ | 扭曲 Edwards 曲线 `Ed25519` 与 Koblitz `secp256k1` | 共识投票 / Solana 跨链内省采用 RFC 8032 `Ed25519`；EVM 账户与 `3-of-4` 守护采用低阶 `secp256k1`（`0x01`） |
| $(R_t, L_t, \rho_t)$ | $R_t, L_t \in \mathbb{N}_0$，$\rho_t = R_t/L_t \geq 100.00\%$ | 协议参考偿付状态模型 HQLA 净值 $R_t$、确定性基准负债状态 $L_t$ 及偿付覆盖率字节码不变量（$\geq 10{,}000$ bps），锚定于 `0x8800000000000000000000000000000000000001` |
| $\mathcal{M}_t^{(\mathrm{LUSD})}$, $\mathcal{M}_t^{(\mathrm{LHKD})}$ | $\mathcal{M}_t \in \mathbb{B}^{32}$（`LINGHE_CAP656_HQLA_V1` 域分离根） | 确定性基准测试储备 Merkle 根：`LUSD` 为 `0x5c3f82332fbffc0be49434232a9a97e2bf2c797423aa3fde4afd19b2c264f410`；`LHKD` 为 `0xa3e5fd1ee51db26128c49f848bc4eb6d71443fbfee51a40b9159832a21beae6e` |
| $\pi_{\mathrm{HKMA}}$ | $\pi_{\mathrm{HKMA}} \in [77500, 78500] \subset \mathbb{Z}_{2^{32}}$ | 香港金管局联系汇率制度标度汇率参数（精度 $10^{-4}$ HKD/USD，中枢 `78000` 即 `7.8000`），由 `0x8900000000000000000000000000000000000000000000000000000000000001` 强制锁定 |
| $A_{24\mathrm{h}}(s,t)$ | $A_{24\mathrm{h}}(s,t) = \sum_{t_k \in (t-86400,\, t]} v_k$ | `FATF Rec. 16` 反拆分 24 小时滑动累计额（单笔法定阈值 US$1,000.00 / HK$7,800.00；24h 滑动累计阈值 US$5,000.00 / HK$39,000.00），超限强制绑定零明文 VASP 信封哈希 |
| $\mathrm{eta}, \Delta_{\mathrm{grace}}$ | $\mathrm{eta} = t_{\mathrm{queue}} + \Delta_{\mathrm{min}}$，$\Delta_{\mathrm{grace}} \in [0, 86400\ \mathrm{s}]$ | 主权治理强制时间锁生效时刻（$\Delta_{\mathrm{min}} \in (0, 30\ \mathrm{days}]$）及跨链双向托管防竞态退款宽限窗（生产配置 `1,800 s`） |

第 01 章 · SECTION 01

# 第一性原理与系统架构动机

## 1.1 自主主权清算不可能三角的形式化建模（The Autonomous Sovereign Settlement Trilemma）

设分布式金融状态机协议族为 $\mathfrak{P}$。对于任意承载自治人工智能体商业与受规管机构级清算的候选协议 $\mathcal{P} \in \mathfrak{P}$，其工程可行性取决于能否在同一原子状态空间内同时满足三项正交的物理与法理硬约束：

1. 亚秒级确定性终局约束（**Sub-Second Deterministic Finality**，$\mathcal{C}_{\mathrm{fin}}$）：交易自进入共识提议至达成不可逆账本提交的端到端延迟上界必须满足 $T_{\mathrm{fin}} \leq 250$ ms，且一旦生成终局证书，在任意拜占庭网络扰动下发生区块重组的概率恒等于零（$P_{\mathrm{reorg}} \equiv 0$），以匹配同步 `HTTP 402` 机器推理调用的毫秒级超时预算。
2. 高并发无阻塞微清算约束（**High-Concurrency Micro-Settlement**，$\mathcal{C}_{\mathrm{conc}}$）：单一机构主权金库或智能体结算协调器在面对 $K \geq 32$ 路并发工作协程以亚美分颗粒度（$0.001–$0.25）高频提交微支付时，必须维持无队头阻塞（Head-of-Line-Free）的线性横向扩展吞吐，且支持直接以零利息法币稳定币原子结算区块执行费，彻底消除预持高波动原生母币的资产负债表摩擦。

3. 内核原生跨法域合规约束（**Kernel-Native Multi-Jurisdictional Compliance**，$\mathcal{C}_{\text{comp}}$）：反洗钱制裁拦截（AML/CFT）、$\geq 100\%$ 破产隔离储备锚定（`Cap. 656`）、法定联系汇率走廊以及 `FATF Recommendation 16` 旅行规则密码学信封存证必须下沉至状态转换内核层，对任意深度的智能合约组合调用（`CALL` / `DELEGATECALL` / 包装衍生凭证）具备数学上的零穿透不可绕过性。

定义协议完备性示性泛函 $\mathcal{J}\colon \mathfrak{P} \to \{0,1\}$ 为三项约束示性函数的逻辑合取：

$$\mathcal{J}(\mathcal{P}) = \mathbb{I}_{\mathcal{C}_{\text{fin}}}(\mathcal{P}) \cdot \mathbb{I}_{\mathcal{C}_{\text{conc}}}(\mathcal{P}) \cdot \mathbb{I}_{\mathcal{C}_{\text{comp}}}(\mathcal{P}) \in \{0,1\}$$

在既有技术范式中，传统央行实时全额结算系统（RTGS）与卡组织网络满足部分合规审查但违背 $\mathcal{C}_{\text{fin}}$ 与 $\mathcal{C}_{\text{conc}}$（固定清算费超越微支付本金且批处理交收滞后达数日）；高速非合规公链或二层排序器试图逼近 $\mathcal{C}_{\text{conc}}$，却牺牲了跨法域主权多中心容错与内核级合规 $\mathcal{C}_{\text{comp}}$；而以太坊等第一代通用智能合约公链则在全部三个维度遭遇底层结构性瓶颈，致使 $\mathcal{J}(\mathcal{P}_{\text{legacy}}) = 0$。

## 1.2 一维串行 EVM、12 秒概率槽位与应用层合规包装器的定量批判

为严格证伪“在传统通用公链之上通过应用层补丁即可支撑自主智能体主权金融”的假设，我们从排队论、共识博弈论与形式化可达性三个维度建立定量批判模型：

第一，一维串行 **Nonce** 的排队论发散与金库级队头阻塞。标准 EVM（`EIP-1559` 类型 `0x02`）强制每个账户地址 $s \in \mathcal{A}$ 维护单一标量计数器 $\text{nonce}(s) \in \mathbb{N}_0$，所有源自 $s$ 的交易必须按全序 $\nu, \nu+1, \nu+2, \dots$ 严格串行提交。当同一企业金库下的多个 AI 智能体协程以聚合泊松到达率 $\lambda$ 并发发起结算时，账户 $s$ 退化为单服务台串行队列 $M/G/1$。设单笔交易经历广播至确认的服务时间为随机变量 $S$（均值 $\mathbb{E}[S] = 1/\mu_1$，二阶矩 $\mathbb{E}[S^2]$），令单通道服务强度 $\rho_1 = \lambda/\mu_1$。由 Pollaczek-Khinchine 公式，稳态平均排队等待时延精确服从：

$$\mathbb{E}[W_q^{(1\text{D})}] = \frac{\lambda \cdot \mathbb{E}[S^2]}{2(1-\rho_1)}, \qquad \lim_{\rho_1 \to 1^-} \mathbb{E}[W_q^{(1\text{D})}] = +\infty$$

一旦序号为 $\nu_0$ 的单笔交易因瞬时网络抖动或基础费率跳变滞留内存池待定区，所有序号 $\nu > \nu_0$ 的后续已签名交易即便携带更高优先费，亦在协议验证入口被判定为序列间隙而陷入全局队头阻塞（HoL Blocking），导致整座机构金库并发吞吐瞬时归零。反之，当灵核OS公链通过二维随机数流形将到达率为 $\lambda$ 的金库流量正交切分为 $K = 32$ 条确定性槽周期服务车道（确定性单槽服务时长 $D = \Delta_{\text{slot}}$，单车道负载 $\rho_K = \lambda D/K$）时，平均排队时延骤降为有界收敛态：

$$\mathbb{E}[W_q^{(K\text{-Lane})}] = \frac{(\lambda/K) \cdot D^2}{2(1-\lambda D/K)}, \qquad \frac{\Theta_{\max}^{(K\text{-Lane})}}{\Theta_{\max}^{(1\text{D})}} = K = 32$$

第二，**12** 秒概率性槽位与多纪元终局的时间风险敞口。采用“提议者—构建者分离 + 双纪元检查点”（如 Gasper PoS）的传统公链以 $\Delta_{\text{slot}}^{(1)} = 12$ s 为出块粒度，而经济终局需跨越 2 epochs = 64 slots $\approx 768$ s（约 12.8 分钟），时间敞口高达灵核OS公链单槽终局周期（250 ms）的 $3,072$ 倍。在长达十余分钟的概率性确认窗口内，区块暴露于重组回滚（$P_{\text{reorg}} > 0$）与最大可提取价值（MEV）三明治夹击风险之下：若智能体网关在零确认或单槽确认阶段提前释放高价值算力结果，攻击者可通过重组双花撤回付款；若网关等待经济终局后再返回推理响应，则数百秒的阻塞延迟直接击穿实时智能体工作流的同步调用闭环。

第三，应用层 **ERC-20** 外挂黑名单的可组合穿透漏洞。现有稳定币普遍依靠应用层 Solidity 合约内的 `mapping(address => bool) blacklisted` 在 `transfer` 函数入口执行检查。设受制裁地址集合为 $\mathcal{B} \subset \mathcal{A}$。若风险主体 $u$ 在被列入 $\mathcal{B}$ 之前（或通过未被及时标记的关联跳板）将代币 $T$ 注入任意无准入校验的包装金库合约或自动做市商池 $W$ 并铸造无权限份额凭证 $W(T)$，则此后 $u$ 向任意对手方 $v$ 转让经济权益的状态转移可形式化表示为：

$$W.\texttt{transfer}(u, v, \delta) \;\Rightarrow\; (\Delta B_{W(T)}(u) = -\delta,\ \Delta B_{W(T)}(v) = +\delta) \;\wedge\; (\Delta B_T(W) \equiv 0)$$

由于状态写入仅发生在外部合约 $W$ 的私有存储槽内，底层代币合约 $T$ 的名义持有人恒定保持为合约地址 $W \notin \mathcal{B}$，从而永不触发 $T.\texttt{transfer}$ 钩子。更严重的是，在跨链意图清算中，源链受制裁主体可借由目标链合规做市商（Solver）向全新净地址垫付资金，彻底切断应用层单跳地址筛查的因果链条。

| 核心架构指标 | 传统电汇 / 卡组织网络 | 第一代串行 **EVM** 公链 | 通用乐观 / **ZK** 二层网络 | **MYRILUM** 灵核**OS**公链（`ChainID: 88888`） |
|---|---|---|---|---|
| 共识与确定性终局时延 | `T+1` 至 `T+3` 工作日批清算 | `~12 s` 出块 / `~768 s` 经济终局 | 中心化排序器软确认 / 跨小时级L1终局 | `250 ms` 单槽确定性终局（$P_{\text{reorg}} \equiv 0$） |

| 核心架构指标 | 传统电汇 / 卡组织网络 | 第一代串行 **EVM** 公链 | 通用乐观 / **ZK** 二层网络 | **MYRILUM** 灵核**OS**公链（`ChainID: 88888`） |
|---|---|---|---|---|
| 同账户并发吞吐模型 | 中心化数据库行锁排队 | `1D Nonce` 串行阻塞（$\rho_1 \to 1^-$ 发散） | 继承 `1D Nonce` 队头阻塞限制 | `0x76` 二维 **Nonce**（默认 **32** 通道，`0` 碰撞） |
| 区块计算带宽与费用媒介 | 固定费 $0.10–$0.30 + 扣率 | `~3.0M Gas/s` · 强制波动母币付 Gas | 受限于 L1 数据可用性成本波动 | `1.2e10 Gas/s` · 原生 `LUSD` / `LHKD` 付 **Gas** |
| 合规拦截与汇率走廊层级 | 事后人工对账与代理行冻结 | 应用层 ERC-20 黑名单（可被包装合约穿透） | 无原生多法域监管原语与联汇边界 | **Rust** 内核 `TIP-403` + `[7.7500, 7.8500]` 硬走廊 |
| 智能体硬件证明与跨链退本 | 不支持机器可验证推理清算 | 模拟 `P-256` 极昂贵 / 多签跨链桥蜜罐 | 缺乏原生 TEE 根证书与双向保护退本 | `0x0100` 低阶 `P-256` + `BN254` 无许可退本金库 |

## 1.3 七层主权架构与 `9+6+2` 合约及预编译全景映射

为从第一性原理彻底消解上述不可能三角，灵核OS公链在不可变创世零块与后创世硬化层构建了垂直正交贯通的七层主权状态机架构，将 **3** 项原生 `TIP-20` 协议资产、**9** 座创世主权系统合约、**6** 项 `Tempo` **Rust** 内核预编译引擎、**2** 组通用与密码学预编译簇以及跨六大外部源链的双向保护托管金库融为单一严密整体（如图 1.1 所示）。




图 **01** — 灵核公链七层主权架构与 **9** 大系统合约 / **6** 大 **Rust** 内核预编译 / **v5.0** 安全守卫全景映射

注：所有 40 位十六进制合约与预编译地址均固化于不可变创世块 01a390d7… 与链配置清单中。

第 02 章 · SECTION 02

# 世界状态空间与全局状态转移总算子

## 2.1 六维正交世界状态元组与不可变创世零块承诺

不同于传统以太坊黄皮书将一切业务状态无差别扁平折叠为单一账户存储字典的设计，灵核OS公链将协议级货币账本、监管策略表、二维并发通道、中央限价订单簿与图灵完备合约存储提升为一等公民，形式化定义任意区块高度 $h \in \mathbb{N}_0$ 的全局世界状态 $\sigma \in \Sigma$ 为六维正交状态元组：

$$\sigma = (\sigma_{\text{acc}}, \sigma_{\text{tip20}}, \sigma_{\text{tip403}}, \sigma_{\text{nonce2d}}, \sigma_{\text{dex}}, \sigma_{\text{sys}}) \in \Sigma$$

六个状态子空间的严格代数签名与物理职责界定如下：

1. 基础账户与原生密钥链子空间 $\sigma_{\text{acc}}$：映射 $\sigma_{\text{acc}} : \mathcal{A} \to \mathbb{Z}_{2^{64}} \times \mathbb{Z}_{2^{256}} \times \mathbb{B}^{32} \times \mathbb{B}^{*} \times \mathcal{K}_{\text{auth}}$，维护 20 字节地址空间 $\mathcal{A} = \mathbb{B}^{20}$ 的传统协议计数器（对应通道 $k = 0$）、基础余额、稀疏存储树根、字节码摘要（含 `0xef` 内核预编译哨兵标识与 `EIP-7702` 委托代码前缀），以及由 `Account Keychain` 预编译（`0xaaaaaaaa0000000000000000000000000000000000`）管理的 WebAuthn / Passkey 授权密钥集。
2. 协议内生 `TIP-20` 货币账本子空间 $\sigma_{\text{tip20}}$：围绕三大 `6` 位小数精度（$10^6$ 原子单位）创世原生资产 $\mathcal{T}_{\text{ens}} = \{\text{LHC}, \text{LUSD}, \text{LHKD}\}$（对应地址 `0x20C0000000000000000000000000000000000000`、`0x20C0000000000000000000000000000000000001` 与 `0x20C0000000000000000000000000000000000002`），维护原子余额映射 $B_\tau : \mathcal{A} \to \mathbb{Z}_{2^{256}}$、流通总供应量 $S_\tau \in \mathbb{Z}_{2^{256}}$、存储槽 `0x02`–`0x04` 内联元数据以及强制绑定的合规策略指针 $\text{policyId}(\tau) \in \mathbb{Z}_{2^{64}}$（其中 `LHKD` 刚性绑定 `Policy #2`）。
3. 内核监管策略注册子空间 $\sigma_{\text{tip403}}$：驻留于 `TIP-403 Policy Registry` 预编译（`0x403c0000000000000000000000000000000000`），定义二元判定算子 $\Phi_{403} : \mathbb{Z}_{2^{64}} \times \mathcal{A} \to \{0, 1\}$，在 Rust 执行器内存热缓存中维护多级机构准入白名单与实时制裁冻结黑名单，支持亚微秒级零字节码解释开销的双向主体合规鉴权。
4. 二维并行随机数流形子空间 $\sigma_{\text{nonce2d}}$：驻留于 `2D Nonce Manager` 预编译（`0x4e4f4e4345000000000000000000000000000000`，高 5 字节即 ASCII `"NONCE"`），定义映射 $\sigma_{\text{nonce2d}} : \mathcal{A} \times \mathbb{Z}_{2^{192}} \to \mathbb{Z}_{2^{64}}$，将每一账户 $s \in \mathcal{A}$ 在任意独立并发通道 $k \in \{0, \ldots, 2^{192} - 1\}$ 下的序列号 $\nu_{s,k} \in \{0, \ldots, 2^{64} - 1\}$ 完全正交解耦。
5. 内核级订单簿与多稳定币费率子空间 $\sigma_{\text{dex}}$：联合 `Enshrined StablecoinDEX` 中央限价订单簿预编译（`0xdec0000000000000000000000000000000000000`，维护 `LUSD/LHC` 与 `LHKD/LHC` 价格—时间优先深度队列）及 `TIP Fee Manager` 预编译（`0xfeec0000000000000000000000000000000000`，维护用户付费代币偏好、验证者本位结算币映射 `validatorTokens`、待提取手续费池 `collectedFees` 与内嵌 `Fee AMM` 流动性储备）。
6. 主权系统合约与治理共识配置子空间 $\sigma_{\text{sys}}$：涵盖创世零块预部署的九大主权系统合约（`0x8004000000000000000000000000000000000001` 至 `0x8004000000000000000000000000000000000003`、`0x8402000000000000000000000000000000000001`、`0x8800000000000000000000000000000000000001`、`0x8900000000000000000000000000000000000001` 至 `0x8900000000000000000000000000000000000004`）、通用授权标准 `Canonical Permit2`（`0x000000000022d473030f116ddee9f6b43ac78ba3`）、`Simplex Validator Config`（`0xcccccccc0000000000000000000000000000000000`）以及后创世审计硬化模块 `LingheSovereignSecurityGuardV5` 的完整持久化状态。

> **定义 2.1 — 主权创世状态不变量**
>
> 灵核OS公链创世初始状态 $\sigma_0 \in \Sigma$ 由网络标识 `ChainID = 88888`（十六进制 `0x15b38`）、自高度 0 即刻激活的 `Osaka` 指令集（`osakaTime = 0`）、运行时区块计算上限 $G_{\max} = 3,000,000,000$ Gas 以及预注入 `Block 0` 的全部原生资产与九大主权合约字节码唯一确定，其规范化创世配置字节流的密码学承诺严格锁定为：
>
> $$\text{SHA-256}(\mathcal{G}_{\text{Linghe}}) \equiv \texttt{96208e9900b218935143be3024d2602566a86d61caa906d42399f35b83afd991}$$
>
> 任意验证节点或主权网关在启动引导阶段均强制执行默认拒绝（`fail-closed`）摘要锁校验：当且仅当本地状态库创世根与 $\text{SHA-256}(\mathcal{G}_{\text{Linghe}})$ 逐比特全等时方可接入共识网络。该创世固化机制在数学上彻底消除了外部账户（EOA）部署者私钥劫持、合约初始化抢跑（Front-running Initialization）以及跨环境寻址漂移风险。

## 2.2 全局区块状态转移总算子与内核预编译分派拓扑

设高度 $h \geq 1$ 的合法提案区块 $B_h = (\mathrm{Hdr}_h, \boldsymbol{T}_h)$ 包含有序交易序列 $\boldsymbol{T}_h = (T_1, T_2, \ldots, T_m)$，其中区块累计消耗计算量受硬边界 $\sum_{i=1}^{m} G_{\mathrm{used}}(T_i) \leq G_{\max} = 3 \times 10^9$ Gas 约束。对于标准 `0x76` 原生并行交易信封：

$$T_i = \texttt{0x76} \,\|\, \mathrm{RLP}([\mathrm{chainId}, \hat{f}_{\mathrm{prio}}, \hat{f}_{\max}, G_{\mathrm{limit}}, \mathrm{calls}_i, \mathscr{L}_{\mathrm{acc}}, k_i, \nu_i, t_{\mathrm{before}}, t_{\mathrm{after}}, t_{\mathrm{fee}}, \sigma_{\mathrm{payer}}, \mathscr{L}_{\mathrm{auth}}, \mathcal{K}^{?}_{\mathrm{auth}}, \sigma_{\mathrm{sender}}])$$

其中批量内部调用向量定义为 $\mathrm{calls}_i = [(c_{i,j}, v_{i,j}, d_{i,j})]_{j=1}^{\ell_i}$。灵核OS公链的区块级全局状态转移总算子 $\Upsilon(\sigma, B_h)$ 形式化表达为单笔交易五阶段原子算子 $\Upsilon_{\mathrm{tx}}$ 的有序复合与块尾提交算子 $\Omega_h$ 的闭包：

$$\Upsilon(\sigma, B_h) = \Omega_h\big((\Upsilon_{\mathrm{tx}}(\cdot, T_m) \circ \Upsilon_{\mathrm{tx}}(\cdot, T_{m-1}) \circ \cdots \circ \Upsilon_{\mathrm{tx}}(\cdot, T_1))(\sigma)\big) \equiv \Omega_h\big(\textstyle\prod_{i=1}^{m} \Upsilon_{\mathrm{tx}}(\cdot, T_i)(\sigma)\big)$$

其中，单笔交易算子分解为五个不可分割的原子子阶段 $\Upsilon_{\mathrm{tx}} = \Upsilon_{\mathrm{accrue}} \circ \Upsilon_{\mathrm{refund}} \circ \Upsilon_{\mathrm{calls}} \circ \Upsilon_{\mathrm{nonce2d}} \circ \Upsilon_{\mathrm{lock}}$：

1. 时间窗核验与前置稳定币锁费（$\Upsilon_{\mathrm{lock}}$）：校验链标识 chainId = 88888、确定性有效时间窗 $t_{\mathrm{after}} < t_{\mathrm{block}} < t_{\mathrm{before}}$ 及发送方签名 $\sigma_{\mathrm{sender}}$（若启用 `0x78` 域分离第三方代付，则同步核验赞助者签名 $\sigma_{\mathrm{payer}}$）。设实际付费主体为 $p \in \{\mathrm{sender}, \mathrm{feePayer}\}$，有效单价 $f_{\mathrm{eff}} = \min(\hat{f}_{\max}, f_{\mathrm{base}} + \hat{f}_{\mathrm{prio}})$，内核从 $\sigma_{\mathrm{tip20}}$ 中原子预扣最大潜在费用 $F_{\max} = G_{\mathrm{limit}} \cdot f_{\mathrm{eff}}$；若 $B_{t_{\mathrm{fee}}}(p) < F_{\max}$，交易在进入执行管线前即刻拒绝。
2. 二维通道序列核验与递增（$\Upsilon_{\mathrm{nonce2d}}$）：向 `0x4e4f4e4345000000000000000000000000000000` 校验 $\sigma_{\mathrm{nonce2d}}(\mathrm{sender}, k_i) = \nu_i$，并原子更新 $\sigma'_{\mathrm{nonce2d}}(\mathrm{sender}, k_i) \leftarrow \nu_i + 1$。对于互异通道键 $k_a \neq k_b$ 的并发交易，只要金库储备充裕且内部写集正交，其状态转换满足阿贝尔交换律，可由 `Tempo Reth` 多核工作窃取线程池无锁并行调度。
3. 哨兵短路路由与原子批量调用（$\Upsilon_{\mathrm{calls}}$）：顺序遍历 $\mathrm{calls}_i$。当目标地址 $c_{i,j}$ 的代码字段为单字节哨兵 `0xef`（遵循 EIP-3541 保留前缀）或命中原生预编译地址段时，分派算子 $\mathcal{D}_{\mathrm{disp}}(c_{i,j}, d_{i,j}, \sigma)$ 直接旁路 EVM 栈式字节码解释循环，零拷贝切入经 LLVM 向量化编译的原生 Rust 内核函数：

$$\mathcal{D}_{\mathrm{disp}}(c, \boldsymbol{d}, \sigma) = \begin{cases} \mathrm{RustKernel}_{\mathrm{TIP20}}(c, \boldsymbol{d}, \sigma_{\mathrm{tip20}}, \sigma_{\mathrm{tip403}}), & \text{if } c \in \{\texttt{0x20C0...0000}, \texttt{0x20C0...0001}, \texttt{0x20C0...0002}\} \\ \mathrm{RustKernel}_{\mathrm{Sys}}(c, \boldsymbol{d}, \sigma_{\mathrm{tip403}}, \sigma_{\mathrm{nonce2d}}, \sigma_{\mathrm{dex}}, \sigma_{\mathrm{acc}}), & \text{if } c \in \{\texttt{0x403c...}, \texttt{0x4e4f...}, \texttt{0xdec0...}, \texttt{0xfeec...}, \texttt{0xaaaa...}, \texttt{0xcccc...}\} \\ \mathrm{CryptoPrecompile}(c, \boldsymbol{d}), & \text{if } c \in \{\texttt{0x01}, \texttt{0x06}, \texttt{0x07}, \texttt{0x08}, \texttt{0x0100}\} \\ \mathrm{RethEVM}_{\mathrm{Osaka}}(c, \boldsymbol{d}, \sigma_{\mathrm{sys}}, \sigma_{\mathrm{acc}}), & \text{otherwise (含创世九大主权合约与通用合约)} \end{cases}$$

任一内部子调用若触发权限回退、算术下溢或 `TIP-403` 合规拦截，整组 $\mathrm{calls}_i$ 的中间写集将被 100% 原子丢弃。表 2.1 给出了全网预编译引擎与创世预部署合约的完整分派拓扑映射。

| 地址空间与精确链上标识 | 模块名称与创世代码形态 | 读写状态子空间 | 执行基底与单次调用计算特征 |
|---|---|---|---|
| `0x20C0000000000000000000000000000000000002` | `LHC` / `LUSD` / `LHKD`（`0xef` 哨兵） | $\sigma_{\mathrm{tip20}}, \sigma_{\mathrm{tip403}}$ | Rust 原生短路函数；零字节码开销，同步内联校验 `TIP-403` |
| `0x403c0000000000000000000000000000000000` | `TIP-403 Policy Registry`（`0xef`） | $\sigma_{\mathrm{tip403}}$ | Rust 内存哈希表直读；亚微秒级准入白名单与制裁黑名单拦截 |
| `0x4e4f4e4345000000000000000000000000000000` | `2D Nonce Manager`（`0xef`） | $\sigma_{\mathrm{nonce2d}}$ | Rust 原生二维通道 $(k, \nu)$ 查询与无锁递增（支持 32+ 并发车道） |
| `0xdec0000000000000000000000000000000000000` | `Enshrined StablecoinDEX`（`0xef`） | $\sigma_{\mathrm{dex}}, \sigma_{\mathrm{tip20}}$ | Rust 内核中央限价订单簿（CLOB），价格—时间优先确定性撮合 |
| `0xfeec0000000000000000000000000000000000` | `TIP Fee Manager`（`0xef`） | $\sigma_{\mathrm{dex}}, \sigma_{\mathrm{tip20}}$ | 多稳定币 Gas 锁费、退费、内嵌 `Fee AMM` 原子换汇与块内防操纵锁 |
| `0xaaaaaaaa00000000000000000000000000000000` | `Account Keychain`（`0xef`） | $\sigma_{\mathrm{acc}}$ | 原生账户授权密钥链与硬件安全隔区 WebAuthn / Passkey 委托 |
| `0xcccccccc00000000000000000000000000000000` | `Simplex Validator Config` | $\sigma_{\mathrm{sys}}$ | 四法域 `Simplex BFT` 验证委员会公钥、权重与结算偏好配置 |

| 地址空间与精确链上标识 | 模块名称与创世代码形态 | 读写状态子空间 | 执行基底与单次调用计算特征 |
|---|---|---|---|
| `0x0000000000000000000000000000000000000006 – 0008` | `BN254 ecAdd / ecMul / ecPairing` | 纯函数无状态 | 原生 `alt_bn128` 曲线运算与四配对检验（实测 Groth16 仅 `815,381 gas`） |
| `0x0000000000000000000000000000000000000100` | `Osaka EIP-7951 P256VERIFY` | 纯函数无状态 | 原生 160 字节输入 `secp256r1` 验签，支撑 H100 / Nitro 硬件根证书 |
| `0x000000000022d473030f116ddee9f6b43ac78ba3` | `Canonical Permit2`（创世字节码） | $\sigma_{\text{sys}}, \sigma_{\text{tip20}}$ | 结构化签名按次有界授权，消除无限 `approve` 残留敞口 |
| `0x8004000000000000000000000000000000000001 – 0003` | `ERC-8004` 身份 / 信誉 / 验证托管 | $\sigma_{\text{sys}}, \sigma_{\text{tip20}}$ | 智能体主权身份、`x402` 履约评分与 `80% / 20%` 零粉尘原子分润 |
| `0x8402000000000000000000000000000000000001` | `LingheSessionKeyManager` | $\sigma_{\text{sys}}$ | 免弹窗智能体会话密钥：单笔上限、总额度、有效期与目标白名单 |
| `0x8800000000000000000000000000000000000001` | `LingheComplianceReserveVault` | $\sigma_{\text{sys}}$ | `Cap. 656` HQLA 储备 Merkle 根与 `FATF Rec. 16` 密码学信封锚定 |
| `0x8900000000000000000000000000000000000001 – 0004` | 全链海关 / ZK 桥 / 硬件验证 / 主权多签 | $\sigma_{\text{sys}}, \sigma_{\text{tip20}}$ | `ERC-7683` 交割、`[77500, 78500]` 走廊、TEE 核验与 `3-of-4` 守护 |

4. 未用算力即时退费与跨币种验证者清分（$\Upsilon_{\text{refund}} \circ \Upsilon_{\text{accrue}}$）：按实际消耗 $G_{\text{used}} \le G_{\text{limit}}$ 退还差额 $(G_{\text{limit}} - G_{\text{used}}) \cdot f_{\text{eff}}$ 至 $p$；若付费币种 $t_{\text{fee}}$ 与当前出块验证者 $v_b$ 在 `0xfeec0000000000000000000000000000000000000000` 注册的本位币 $t_v$ 异构，内嵌 `Fee AMM` 在同一原子上下文内自动完成跨稳定币兑付并记入 `collectedFees[v_b][t_v]`。
5. 块尾稀疏状态树承诺（$\Omega_h$）：块内全部交易执行完毕后，$\Omega_h$ 驱动并行稀疏 Merkle 前缀树并发重算脏子树哈希，生成唯一确定性状态根 $\mathcal{R}(\sigma_h)$ 并由零拷贝内存映射存储引擎异步持久化。

**不变量 2.1 — 原生资产与策略内生耦合律**

对任意协议内生 `TIP-20` 资产 $\tau \in \mathcal{T}_{\text{ens}}$（设其绑定策略标识为 $\psi_\tau = \text{policyId}(\tau)$）及任意世界状态转移轨迹 $\sigma \xrightarrow{T} \sigma'$，设账户 $a \in \mathcal{A}$ 的原子余额增量为 $\Delta B_\tau(a) = B'_\tau(a) - B_\tau(a)$。无论调用路径源自顶层 `0x76` 交易、智能合约多跳嵌套 `CALL` / `DELEGATECALL`、第三方 `0x78` 代付赞助、抑或内核订单簿撮合，余额发生非零变动的充要前提是在 Rust 内核分派入口同步满足双向策略守卫谓词：

$$\forall \tau \in \mathcal{T}_{\text{ens}},\ \forall (s, r) \in \mathcal{A}^2 :\quad (\Delta B_\tau(s) < 0 \ \vee\ \Delta B_\tau(r) > 0) \ \Rightarrow\ (\psi_\tau \le 1 \ \vee\ (\Phi_{403}(\psi_\tau, s) = 1 \ \wedge\ \Phi_{403}(\psi_\tau, r) = 1))$$

一旦资金实际发送方 $s$ 或实际接收方 $r$ 任一方被列入 `0x403c0000000000000000000000000000000000000000` 制裁黑名单（使得 $\Phi_{403}(\psi_\tau, \cdot) = 0$），底层 Rust 状态函数立即触发硬回滚，任何外层包装器合约或干净的 Gas 赞助人均无法在信息论与状态可达性意义上绕过该内核耦合律。

第 03 章 · SECTION 03

# Commonware Simplex BFT 250ms 流水线共识与跨法域容错证明

## 3.1 四法域主权验证委员会拓扑与部分同步对抗网络模型

为在单一国家或单一云区域遭遇极端物理断网、海底光缆切断或局部数据中心隔离维护时维持分布式状态机连续性，灵核OS公链拒绝单数据中心排序器模型，将共识验证委员会定义为横跨亚太与欧洲四大国际金融中心高等级数据中心的有权有界集合 $\mathcal{V} = \{v_0, v_1, v_2, v_3\}$。系统节点总数 $n = |\mathcal{V}| = 3f + 1 = 4$，最大可容忍拜占庭恶意节点数 $f = 1$，法定仲裁人数阈值（Quorum Size）严格设定为：

$$q = \lceil \frac{2n+1}{3} \rceil = 2f + 1 = 3$$

委员会成员的 `Ed25519` 共识公钥、权重及费用本位币注册于内核预编译合约 `Simplex Validator Config`（`0xcccccccc00000000000000000000000000000000`），并与主权治理时间锁多签金库 `LingheSovereignTimelockMultisig`（`0x8900000000000000000000000000000000000004`）保持严格同构的 `3-of-4` 跨法域门限拓扑（表 3.1）。各节点签名私钥均封装于硬件密钥管理服务（KMS）保护的加密信封内，杜绝明文密钥落盘。

| 节点索引 | 验证节点标识 | 主权监管辖区与合规对标基准 | 共识职责定位 | 验证者 32 字节 `Ed25519` 公钥承诺 | 本位结算资产（`feeToken`） |
|---|---|---|---|---|---|
| $v_0$ | `hk-validator-01` | 中国香港（`HK-HKMA · Cap. 656`） | 主提议与合规基准验证节点 | `0x02b4c08cb8ccfa85f4f2c097ea329be89f3f47682814ba8e9f9b2e7daf9327c9` | `LHKD`（`0x20C000000000000000000000000000000000002`） |
| $v_1$ | `sg-validator-02` | 新加坡（`SG-MAS · Project Guardian`） | 法定仲裁验证节点 | `0x7fb08589ecc82c26d5903050333c542350bf0edf71007862d146681bf31193b6` | `LUSD`（`0x20C000000000000000000000000000000000001`） |
| $v_2$ | `tyo-validator-03` | 日本东京（`JP-FSA · 资金决济法`） | 法定仲裁验证节点 | `0xada429359142ec4d332a5283fb27e28145c552427a65cf464f44137e091345f4` | `LUSD`（`0x20C000000000000000000000000000000000001`） |
| $v_3$ | `zrh-validator-04` | 瑞士苏黎世（`CH-FINMA · DLT 法案`） | 异洲仲裁与归档见证节点 | `0xccd374991c362b1dee56639fc1ad1da4adc0bb5f2262ce120147077eebacec5a` | `LHC`（`0x20C000000000000000000000000000000000000`） |

对抗网络与威胁模型：底层点对点通信遵循 Dwork-Lynch-Stockmeyer 部分同步模型（Partial Synchrony）。设存在未知全局稳定时间 $\mathrm{GST} \geq 0$ 及已知的诚实节点间单向消息传播延迟上界 $\Delta_{\text{net}} > 0$：

1. 异步风暴期（$t < \mathrm{GST}$）：受制于跨洋光缆抖动、分布式拒绝服务（DDoS）冲击或自治系统边界网关协议（BGP）路由劫持，网络消息可遭遇任意有限延迟、乱序或拓扑分区；
2. 同步稳态期（$t \geq \mathrm{GST}$）：任意两诚实验证节点 $v_i, v_j \in \mathcal{V}_{\text{honest}}$ 之间发出的共识报文必在 $\Delta_{\text{net}}$ 时间内可靠送达；
3. 自适应拜占庭敌手 $\mathcal{A}$：多项式计算有界的敌手 $\mathcal{A}$（无法伪造诚实节点的 `Ed25519` 签名或碰撞 `SHA-256` / `Keccak-256` 哈希）最多可完全腐化并协同控制 $|\mathcal{V}_{\text{byz}}| \leq f = 1 < n/3$ 名验证节点，使其实施静默停机、向不同司法辖区节点广播冲突双签提案（Equivocation）或刻意在超时临界点释放报文。诚实节点集合满足 $|\mathcal{V}_{\text{honest}}| = n - f \geq 3 = q$。

### 3.2 三阶段流水线槽位动力学与虚拟空块无停顿跳槽机制

`Commonware Simplex BFT` 将逻辑时间划分为严格单调递增的槽位视图（Slot / View）$h \in \mathbb{N}_1$，单槽目标周期 $\Delta_{\text{slot}} = 250\ \text{ms}$。每个槽位 $h$ 由确定性轮转函数 $\mathcal{L}(h) = v_{(h \bmod n)}$ 指定唯一主提议者，在单一槽位窗口内执行全互联（All-to-All）三阶段确定性流水线，无需传统 PBFT / HotStuff 复杂的三轮串行领导者收集或指数退避视图切换（View-Change）：

1. 第一阶段：状态预取提案广播（**Stage 1 — `PROPOSE`**）。一旦进入槽位 $h$（即观测到前序槽位 $h-1$ 的有效公证证书 $\mathcal{C}_{\text{not}}(h-1, B_{h-1})$ 或空块跳过证书 $\mathcal{C}_{\text{dummy}}(h-1)$），当值提议者 $\mathcal{L}(h)$ 从专用支付车道与通用内存池提取无冲突 `0x76` 交易集，构造以父块状态根为前驱的候选区块 $B_h$ 连同热存储槽预取提示（Execution Prefetch Hints），向全网广播签名提案 $\langle \text{PROPOSE}, h, B_h, \mathcal{C}_{\text{prev}} \rangle_{\sigma_{\mathcal{L}(h)}}$。
2. 第二阶段：并行状态核验与公证投票（**Stage 2 — `NOTARIZE`**）。诚实验证节点 $v_i \in \mathcal{V}_{\text{honest}}$ 在本地槽定时器 $T_{\text{dummy}}$ 到期前，仅对槽位 $h$ 收到的首个因果合法且通过 `Tempo Reth` 并行状态转移校验（即 $\Upsilon(\sigma_{h-1}, B_h)$ 计算状态根与区块头声明完全一致）的提案，全网广播唯一公证投票 $\langle \text{NOTARIZE}, h, H(B_h) \rangle_{\sigma_i}$。任意节点集齐来自 $q = 3$ 名互异验证者对同一摘要 $H(B_h)$ 的有效公证签名，即合成槽位公证证书（**Notarization Certificate**）：
$$\mathcal{C}_{\text{not}}(h, B_h) = \{ \langle \text{NOTARIZE}, h, H(B_h) \rangle_{\sigma_i} \mid v_i \in Q_{\text{not}},\ |Q_{\text{not}}| = q = 3 \}$$
同时，$\mathcal{C}_{\text{not}}(h, B_h)$ 的生成即刻解锁下一槽位 $h+1$ 提议者 $\mathcal{L}(h+1)$ 的乐观流水线出块权限，实现跨槽位零空闲重叠。
3. 第三阶段：不可逆终局提交（**Stage 3 — `FINALIZE`**）。若诚实节点 $v_i$ 在槽位 $h$ 已观测到 $\mathcal{C}_{\text{not}}(h, B_h)$ 且在此之前未签署过槽位 $h$ 的超时跳过票，则立即全网广播终局投票 $\langle \text{FINALIZE}, h, H(B_h) \rangle_{\sigma_i}$。一旦集齐 $q = 3$ 份互异验证者的终局签名，即合成单槽确定性终局证书（**Finality Certificate**）：
$$\mathcal{C}_{\text{fin}}(h, B_h) = \{ \langle \text{FINALIZE}, h, H(B_h) \rangle_{\sigma_j} \mid v_j \in Q_{\text{fin}},\ |Q_{\text{fin}}| = q = 3 \}$$
此时区块 $B_h$ 及其中包含的全部 `x402` 微支付、`TIP-403` 合规账本与跨链释放状态达成永久不可逆终局。
4. 并行虚拟空块跳槽旁路（**Dummy Slot Skip**）：每个诚实节点在进入槽位 $h$ 时独立启动有界计时器 $T_{\text{dummy}}$。若因 $\mathcal{L}(h)$ 拜占庭静默或瞬时链路抖动导致在 $T_{\text{dummy}}$ 内未能合成公证证书且未投出 `FINALIZE` 票，诚实节点直接广播虚拟空块投票 $\langle \text{DUMMY}, h, \perp \rangle_{\sigma_i}$；集齐 $q = 3$ 份签名即形成跳过证书 $\mathcal{C}_{\text{dummy}}(h)$，以 $O(n^2)$ 常数轮通信直接推进至槽位 $h+1$（如图 3.1 所示）。

与经典 PBFT 在领导者故障时需收集包含历史准备集证明的 `VIEW-CHANGE` 向量并广播大型 `NEW-VIEW` 报文（消息比特复杂度高达 $O(n^3)$ 且引发多轮超时倍增停顿）截然不同，`Simplex BFT` 将正常出块与异常跳槽统一收敛为对称的单轮全互联投票原语。设单槽提案载荷字节数为 $|B_h|$，单枚 `Ed25519` 签名为 64 bytes，则无论处于稳态终局还是拜占庭跳槽工况，单槽全网总通信比特复杂度恒定服从紧致上界：

$$\mathcal{M}_{\text{slot}}(n) = (n-1) \cdot |B_h| + 2n(n-1) \cdot |\sigma_{\text{Ed25519}}| = O(n|B_h| + n^2)$$

在 $n = 4$ 的四法域主权拓扑下，$2n(n-1) = 24$ 次轻量级签名交换仅产生 1.5 KB 控制面开销，使跨洲专线带宽几乎 100% 服务于高吞吐交易有效载荷传输；同时，槽位 $h+1$ 的提案组装在观测到 $\mathcal{C}_{\text{not}}(h, B_h)$ 的瞬间即与槽位 $h$ 的第三阶段 `FINALIZE` 广播在时间轴上流水线重叠执行，彻底消除槽间空闲泡沫。

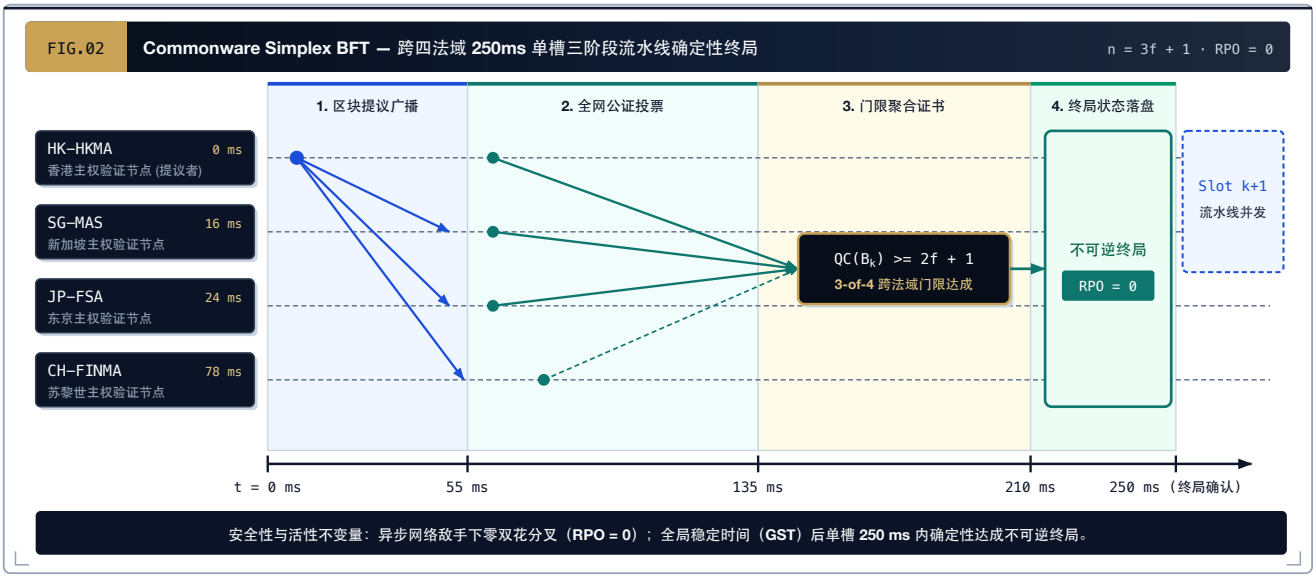



**图 02 — Commonware Simplex BFT 单槽 250ms 流水线共识与跨四法域门限终局时序图**

注：槽位 k+1 区块提议在槽位 k 进入公证聚合后即重叠启动，消除传统多轮 BFT 空置等待。

### 3.3 可问责安全性（Accountable Safety）与有界亚秒级活性形式化证明

本节严格证明 `Commonware Simplex BFT` 在四法域 `3-of-4` 拓扑下同时具备全异步无条件零分叉安全性（含超阈值腐化下的密码学可问责追责）与同步稳态期亚秒级有界活性。

**定理 3.1 — 单槽确定性终局与双花不可能性（Accountable Safety）**

设验证者委员会规模 $n = 4$，法定人数阈值 $q = 3$，拜占庭恶意节点数满足 $|\mathcal{V}_{\text{byz}}| \le f = 1$。无论网络处于异步风暴期（$t < \text{GST}$）还是同步稳态期（$t \ge \text{GST}$），对任意槽位 $h, h' \in \mathbb{N}_1$ 及任意两诚实节点 $v_a, v_b \in \mathcal{V}_{\text{honest}}$：

1. 同槽唯一终局性：若 $\mathcal{C}_{\text{fin}}(h, B)$ 与 $\mathcal{C}_{\text{fin}}(h, B')$ 均有效，则必有 $H(B) = H(B')$；
2. 跨槽因果不可逆性：若区块 $B$ 在槽位 $h$ 获得终局证书 $\mathcal{C}_{\text{fin}}(h, B)$，则任何后续槽位 $h' > h$ 获得公证或终局证书的区块 $B'$ 必以 $B$ 为严格确定的祖先区块，即已终局交易的重组与双花概率恒满足 $P_{\text{reorg}} \equiv 0$；
3. 密码学可问责性（**Accountable Attribution**）：倘若极端违反容错假设（$|\mathcal{V}_{\text{byz}}| > f$）导致出现两份冲突终局证书，协议可在 $O(q)$ 时间内提取出至少 $2q - n = 2$ 名恶意双签验证者的不可抵赖密码学罪证（Slashing Evidence）。

**证明**

第一步（同槽冲突证书不可能性与交集下界推导）：采用反证法。假设在同一槽位 $h$ 存在两个互异区块 $B \neq B'$（即 $H(B) \neq H(B')$）分别获得了有效公证或终局证书，记其对应的签名验证者集合分别为 $Q_1 \subseteq \mathcal{V}$ 与 $Q_2 \subseteq \mathcal{V}$。由证书合法性定义，$|Q_1| \ge q = 3$ 且 $|Q_2| \ge q = 3$。因 $Q_1 \cup Q_2 \subseteq \mathcal{V}$ 故 $|Q_1 \cup Q_2| \le n = 4$。应用有限集容斥原理（Inclusion-Exclusion Principle），两法定仲裁集的交集基数满足严格下界：

$$|Q_1 \cap Q_2| = |Q_1| + |Q_2| - |Q_1 \cup Q_2| \ge q + q - n = 2q - n = 6 - 4 = 2 > f = 1$$

鉴于拜占庭敌手至多控制 $|\mathcal{V}_{\text{byz}}| \le f = 1$ 名节点，从交集中剔除全部潜在拜占庭节点后，剩余诚实节点子集的基数满足：

$$|(Q_1 \cap Q_2) \setminus \mathcal{V}_{\text{byz}}| \ge |Q_1 \cap Q_2| - |\mathcal{V}_{\text{byz}}| \ge 2 - 1 = 1$$

由此推出：必存在至少一名诚实验证者 $v^* \in \mathcal{V}_{\text{honest}}$ 在同一槽位 $h$ 对两个不同区块摘要 $H(B) \neq H(B')$ 签发了 `NOTARIZE`（进而触发 `FINALIZE`）签名。然而，按照 3.2 节诚实节点状态机规约，诚实节点在每个槽位 $h$ 有且仅对首个校验通过的提案投出唯一一次 `NOTARIZE` 票，严格禁止同槽双签，产生逻辑矛盾。故同槽绝无可能形成两份冲突的公证或终

局证书。同时，若敌手突破 $f=1$ 假设强行构造分叉，直接求交集 $Q_1 \cap Q_2$ 即可公开捕获至少 2 对带 `Ed25519` 签名的冲突报文 $(\langle \cdot, h, H(B) \rangle_{\sigma_k}, \langle \cdot, h, H(B') \rangle_{\sigma_k})$，实现确定性链上罚没定责。

第二步（跨槽锁定与空块跳过互斥性归纳证明）：现设区块 $B$ 在槽位 $h$ 已生成有效终局证书 $\mathcal{C}_{\text{fin}}(h, B)$，其签名者集合为 $Q_{\text{fin}}$（$|Q_{\text{fin}}| = 3$）。我们对后续槽位 $h' > h$ 施行数学归纳。考察槽位 $h$ 是否可能同时生成虚拟空块跳过证书 $\mathcal{C}_{\text{dummy}}(h)$：由于 $Q_{\text{fin}}$ 中至少包含 $|Q_{\text{fin}} \setminus \mathcal{V}_{\text{byz}}| \geq q - f = 3 - 1 = 2$ 名诚实验证者，根据阶段 3 与阶段 4 的互斥状态转移规则，任何已广播 $\langle \text{FINALIZE}, h, H(B) \rangle$ 的诚实节点在该槽位已永久关闭 `DUMMY` 投票分支。因此，全网在槽位 $h$ 最多只能由未参与 $Q_{\text{fin}}$ 的 $(n-q) = 1$ 名节点及至多 $f = 1$ 名拜占庭双面节点投出跳过票，总票数上界为：

$$N_{\text{dummy}}^{\max}(h) \leq (n-q) + f = (4-3) + 1 = 2 < q = 3$$

故槽位 $h$ 绝对无法凑齐生成 $\mathcal{C}_{\text{dummy}}(h)$ 所需的 3 份签名。结合第一步结论（槽位 $h$ 不存在异于 $B$ 的另一公证证书 $\mathcal{C}_{\text{not}}(h, B')$），槽位 $h+1$ 的任何合法提案欲通过诚实节点的父块证书校验，其唯一合法的前驱公证引用必为 $\mathcal{C}_{\text{not}}(h, B)$（或者若槽位 $h+1$ 自身被 `DUMMY` 跳过，该唯一非空祖先指针无损传递至 $h+2$）。依归纳法，对一切 $h' > h$，所有后续上链区块必位于以 $B$ 为根的唯一确定性链条之上，$P_{\text{reorg}} \equiv 0$。证毕。

Q.E.D. ■

**定理 3.2 — GST 后有界亚秒级活性上界（Bounded Liveness）**

在部分同步网络进入同步窗口 $t \geq \text{GST}$ 后，设存活诚实法定人数集群内部单向网络传播上界为 $\Delta_{\text{quorum}}$（在单一异洲辖区如苏黎世节点完全断网的极端工况下，由香港 `hk-validator-01`、新加坡 `sg-validator-02` 与东京 `tyo-validator-03` 组成的亚太光纤法定人数满足 $\Delta_{\text{APAC}} \leq 35$ ms），提案组装耗时上界为 $\tau_{\text{prop}}$，单块 `Tempo Reth` 并行执行耗时上界为 $\tau_{\text{exec}}$，多核 `Ed25519` 批量验签开销为 $\tau_{\text{crypto}}$，本地空块超时阈值为 $T_{\text{dummy}} = \Delta_{\text{slot}} = 250$ ms。则：

1. 当轮值提议者 $\mathcal{L}(h) \in \mathcal{V}_{\text{honest}}$ 为在线诚实节点时，新区块必在单槽上界 $T_{\text{finality}} \leq \tau_{\text{prop}} + \tau_{\text{exec}} + 3\Delta_{\text{quorum}} + \tau_{\text{crypto}} < \Delta_{\text{slot}} = 250$ ms 内达成不可逆终局；
2. 即使当前槽位恰好轮值至拜占庭故障或跨洲断网节点 $\mathcal{L}(h) \in \mathcal{V}_{\text{byz}}$，系统亦必在严格有界时间 $T_{\text{recover}} \leq T_{\text{dummy}} + \Delta_{\text{quorum}} + T_{\text{finality}} < 500$ ms（至多两个槽位周期）内自愈并完成下一诚实区块终局确认。

证明

在 $t \geq \text{GST}$ 且至多存在 $f = 1$ 名故障或分区离线节点的条件下，存活诚实节点子集 $\mathcal{V}_{\text{live}} \subseteq \mathcal{V}_{\text{honest}}$ 的规模满足 $|\mathcal{V}_{\text{live}}| \geq n - f = 3 = q$，单凭 $\mathcal{V}_{\text{live}}$ 自身即可独立闭合公证、终局与跳槽三类法定人数仲裁，绝不因单点或单法域失联而发生死锁。分两种完备互斥情形讨论：

情形一（诚实提议者轮值槽位，$\mathcal{L}(h) \in \mathcal{V}_{\text{live}}$）：设槽位 $h$ 于时刻 $t_0 \geq \text{GST}$ 开启。提议者 $\mathcal{L}(h)$ 在 $\tau_{\text{prop}}$ 内完成交易打包并向 $\mathcal{V}_{\text{live}}$ 广播提案，所有存活诚实节点至迟于 $t_1 \leq t_0 + \tau_{\text{prop}} + \Delta_{\text{quorum}}$ 完整接收 $B_h$。随后各节点经 `Tempo Reth` 并行管线在 $\tau_{\text{exec}}$ 内核验状态根无误，立刻全互联广播 `NOTARIZE` 票；至迟于 $t_2 \leq t_1 + \tau_{\text{exec}} + \Delta_{\text{quorum}} + \tau_{\text{crypto}}/2$，全部 3 名存活诚实节点集齐 $q = 3$ 份签名合成 $\mathcal{C}_{\text{not}}(h, B_h)$。因 $t_2 - t_0 < T_{\text{dummy}}$，无任何诚实节点触发空块跳过票，三节点随即全互联广播 `FINALIZE` 票，至迟于 $t_3 \leq t_2 + \Delta_{\text{quorum}} + \tau_{\text{crypto}}/2$ 全部合成终局证书 $\mathcal{C}_{\text{fin}}(h, B_h)$。代入亚太三中心专线实测工程参数（$\tau_{\text{prop}} \approx 15$ ms，$\tau_{\text{exec}} \approx 45$ ms，$\Delta_{\text{APAC}} \approx 32$ ms，$\tau_{\text{crypto}} \approx 4$ ms），单槽确定性终局总时延上界为：

$$T_{\text{finality}} = t_3 - t_0 \leq \tau_{\text{prop}} + \tau_{\text{exec}} + 3\Delta_{\text{APAC}} + \tau_{\text{crypto}} \approx 15 + 45 + 3 \times 32 + 4 = 160 \text{ ms} < \Delta_{\text{slot}} = 250 \text{ ms}$$

情形二（拜占庭恶意或断网提议者轮值槽位，$\mathcal{L}(h) \notin \mathcal{V}_{\text{live}}$）：若 $\mathcal{L}(h)$ 恶意扣块静默或释放非法区块，$\mathcal{V}_{\text{live}}$ 中的 3 名诚实节点至迟在 $t_0 + T_{\text{dummy}}$ 同步触发本地槽定时器并全互联广播 $\langle \text{DUMMY}, h, \perp \rangle$，在 $\Delta_{\text{quorum}}$ 内即刻聚合成跳过证书 $\mathcal{C}_{\text{dummy}}(h)$ 并无阻塞推进至槽位 $h+1$（若拜占庭提议者在临界点仅向部分诚实节点投递合法提案使某节点合成了 $\mathcal{C}_{\text{not}}(h, B_h)$，则该公证证书随广播在 $\Delta_{\text{quorum}}$ 内同步送达其余诚实节点，同样解除阻塞进入槽位 $h+1$）。由于 $|\mathcal{V}_{\text{byz}}| \leq f = 1 < n = 4$，确定性模 4 轮转算子 $\mathcal{L}(h+1) = v_{((h+1) \bmod 4)}$ 必满足 $\mathcal{L}(h+1) \neq \mathcal{L}(h)$，故下一槽位提议者 100% 落入在线诚实集合 $\mathcal{V}_{\text{live}}$，随即复归情形一，在单槽 $T_{\text{finality}}$ 内达成确定性终局。两阶段总恢复时延严格有界：

$$T_{\text{recover}} \leq T_{\text{dummy}} + \Delta_{\text{APAC}} + T_{\text{finality}} \approx 250 + 32 + 160 = 442 \text{ ms} < 2\Delta_{\text{slot}} = 500 \text{ ms}$$

由此证明协议在任意单法域断网或单拜占庭节点作恶下永不发生级联停机，恒定维持亚秒级确定性活性。证毕。

Q.E.D. ■

### 3.4 复合对抗边界分析：跨洲海缆中断叠加提案双签与抢跑操纵防御

在真实主权金融对抗环境中，国家级或有组织敌手往往实施网络层与共识层的多向量协同攻击。结合定理 3.1 与定理 3.2，灵核OS公链在以下三类极端复合工况下均保持确定性免疫：

1. 亚欧海缆切断与恶意提议者双签叠加攻击（**Partition + Equivocation**）：设亚欧跨洋海缆物理中断导致苏黎世节点 $v_3$ 暂时孤立，与此同时被腐化的亚太节点（假设为 $v_2$）在其轮值槽位向香港 $v_0$ 发送候选块 $B^{(A)}$、向新加坡 $v_1$ 发送包含冲突双花交易的候选块 $B^{(B)}$。此时诚实节点 $v_0$ 仅签署 $H(B^{(A)})$，$v_1$ 仅签署 $H(B^{(B)})$；即使恶意节点 $v_2$ 同时为两个分支双签背书，$B^{(A)}$ 与 $B^{(B)}$ 各自最多仅能收集到 $1$（诚实）$+1$（拜占庭）$=2<q=3$ 份签名，均无法跨过公证门槛。当 $250$ ms 槽定时器触发后，$v_0, v_1$（及恢复连接后的 $v_3$，或由 $v_2$ 之外诚实节点轮值的后续槽位）安全推进视图，且 $v_2$ 发出的两份同槽异构提案签名直接构成可验证的密码学双签罚没证据。
2. 块内费率代币切换套利封堵（**Intra-Block Fee Token Manipulation**）：若拜占庭提议者试图根据待打包 `0x76` 交易流的稳定币分布，在同一区块内动态反复调用 `0xfeec0000000000000000000000000000000000000000` 修改自身 `validatorToken` 以套取内嵌 `Fee AMM` 价差，底层 Rust 预编译强制施加块级状态锁（触发 `CannotChangeWithinBlock` 硬回退），将提议者的单槽费率操纵空间精确压缩为零。
3. 单法域司法强制或单点密钥泄露隔离：由于共识公证（`0xcccccccc0000000000000000000000000000000000`）与治理多签（`0x8900000000000000000000000000000000000004`）均采用横跨 `HK-HKMA`、`SG-MAS`、`JP-FSA`、`CH-FINMA` 四大独立法域的 `3-of-4` 门限架构，且叠加 `LingheSovereignSecurityGuardV5` 的强制链上时间锁延迟 $\mathrm{eta} = t_{\mathrm{queue}} + \Delta_{\min}$（未到期执行确定性回滚 `TIMELOCK_ETA_NOT_REACHED`），任何单一司法辖区的突发行政指令或最多 $f=1$ 把守护者密钥妥协，既无法审查冻结全网区块生产，亦无法单方面篡改协议参数或触碰隔离储备资产。

---

第 04 章 · SECTION 04

## Tempo Reth 并行执行引擎与 0x76 二维随机数无锁代数

亚秒级拜占庭共识仅解决了跨司法辖区区块提议与终局投票的通信延迟下界；若底层虚拟机仍沿用传统单线程栈式解释执行与一维串行账户序列号（1D Sequential Nonce），高频自治智能体集群与机构清算网关将在状态机入口遭遇严重的队头阻塞（Head-of-Line Blocking, HoL）与磁盘状态树 I/O 放大。灵核OS公链（`Linghe Chain`，`ChainID: 88888` / `0x15b38`）在执行层深度定制基于 Rust 构建的模块化执行引擎 `Tempo Reth`（原生激活大阪升级指令集），将单区块计算上限提升至 $G_{\max} = 3,000,000,000$ Gas，在单槽确定性终局周期 $\Delta_{\mathrm{slot}} = 250$ ms 下提供持续理论计算带宽：

$$\Theta_{\mathrm{gas}} = \frac{G_{\max}}{\Delta_{\mathrm{slot}}} = \frac{3,000,000,000 \text{ Gas}}{0.25 \text{ s}} = 1.2 \times 10^{10} \text{ Gas/s}$$

这一较传统主网单线程执行模型（约 $3.0 \times 10^{6}$ Gas/s）达 $4,000$ 倍的计算吞吐跃升，建立在原生 **Rust** 哨兵短路预编译、`0x76` 二维并行随机数流形以及协议内生稳定币费用抽象三大正交代数机制之上。

### 4.1 模块化 Rust 零拷贝执行管线与二维随机数状态流形

在 `Tempo Reth` 内部，状态转换函数 $\Upsilon(\Sigma_h, B_{h+1}) \to \Sigma_{h+1}$ 通过三重系统级工程架构消除解释器与存储瓶颈：

1. `EIP-3541` `0xef` 哨兵字节码短路路由（**Sentinel Short-Circuiting**）：创世第 `0` 块（不可变创世摘要 `96208e9900b218935143be3024d2602566a86d61caa906d42399f35b83afd991`）将三大原生 `TIP-20` 资产及核心状态预编译账户的代码段置为保留单字节哨兵 `0xef`。当执行调度器识别目标调用地址携带 `0xef` 哨兵或命中内核预编译地址段时，直接旁路 EVM 操作码分派循环与 256 位栈帧拷贝，零拷贝切入经 LLVM 向量化编译的原生 Rust 状态机函数，将单笔合规稳定币转账及策略核验的 CPU 时钟开销压缩至亚微秒量级。
2. 内存态热缓存与多核稀疏默克尔前缀树（**Parallel Sparse Merkle-Patricia Trie**）：引擎在常驻内存中维护热账户与存储槽快照，区块执行完毕后利用工作窃取（Work-Stealing）多核线程池并发重算脏子树状态根，底层持久化委托零拷贝内存映射引擎 `MDBX` 异步落盘，彻底根除传统客户端块尾同步刷新 B+ 树引发的随机写放大。
3. 专用支付车道区块空间隔离（**Dedicated Payment Lane Isolation**）：交易池与区块构建器将高频 `TIP-20` 稳定币结算流同任意复杂度的通用智能合约计算实施区块配额隔离，确保外部高复杂度合约调用的瞬时 Gas 峰值绝不挤压智能体微支付与机构法币清算的确定性上链窗口。

在账户并发控制维度，传统 EVM 强制要求每个账户 $s \in \mathcal{A}$ 维护单一标量计数器 $\mathrm{nonce}(s) \in \mathbb{N}_0$。当 $K$ 个独立智能体工作协程共享同一企业金库并以聚合泊松到达率 $\lambda$ 提交微支付请求时，该账户退化为单服务台串行排队系统 $M/G/1$。设单笔交易经历签名、广播至单槽终局确认的服务时间随机变量为 $S$（均值 $\mathbb{E}[S] = 1/\mu_1$，二阶矩 $\mathbb{E}[S^2]$，服务强度 $\rho_1 = \lambda/\mu_1$），由 Pollaczek-Khinchine 排队论方程可知稳态期望排队等待时延为：

$$\mathbb{E}[W_q^{(1\mathrm{D})}] = \frac{\lambda \cdot \mathbb{E}[S^2]}{2(1-\rho_1)}, \qquad \lim_{\rho_1 \to 1^-} \mathbb{E}[W_q^{(1\mathrm{D})}] = +\infty$$

一旦序号为 $\nu_0$ 的单笔交易因网络抖动滞留于待定池，后续全部序号 $\nu > \nu_0$ 的合法已签名交易均被协议层死锁阻塞。为从代数根源上解除非相关业务流之间的虚假全序耦合，`Linghe Chain` 通过内核预编译 `2D Nonce Manager`（`0x4e4f4e4345000000000000000000000000000000`，高 5 字节 `0x4e4f4e4345` 对应 ASCII `"NONCE"`）将重放保护空间从一维半群射线升维至二维离散乘积流形：

$$\mathcal{N} : \mathcal{A} \times \{0, 1, \ldots, 2^{192}-1\} \to \{0, 1, \ldots, 2^{64}-1\}, \qquad (s, k) \mapsto \nu_{s,k}$$

其中，第一维分量 $k \in \{0, \ldots, 2^{192}-1\}$ 为 192 位无符号通道键（`nonceKey`），第二维分量 $\nu_{s,k} \in \{0, \ldots, 2^{64}-1\}$ 为账户 $s$ 在通道 $k$ 内部的单调递增序列号（`nonceSequence`）。通道 $k = 0$ 严格映射至经典以太坊账户状态中的一维协议随机数槽，保障存量 `EIP-1559`（`0x02`）工具链零修改兼容；通道域 $k \in \{1, \ldots, 2^{192}-1\}$ 则由预编译 `0x4e4f4e4345000000000000000000000000000000` 在独立内核存储命名空间中管理，对外暴露标准确定性只读接口 `getNonce(address account, uint256 nonceKey) external view returns (uint64)`，并在每次状态跃迁后发射 `NonceIncremented(address indexed account, uint256 indexed nonceKey, uint64 newNonce)` 事件。

**定义 4.1 — 正交二维随机数通道（Orthogonal 2D Nonce Lane）**

给定主权账户或智能体清算协调器地址 $s \in \mathcal{A}$，其在通道键 $k \in \{1, \ldots, 2^{192}-1\}$ 上的正交二维随机数通道定义为受控于二元状态坐标 $(s, k)$ 的独立单调状态纤维 $\mathcal{F}_{s,k} = \{(s, k, \nu) \mid \nu \in \mathbb{Z}_{2^{64}}\}$。任意两笔携带互异通道键 $k_i \neq k_j$ 的 `0x76` 交易 $T_i, T_j$，在 `0x4e4f4e4345000000000000000000000000000000` 内核状态树中的读写投影集 $\mathfrak{RW}_{\mathrm{nonce}}(T_i) = \{(s, k_i)\}$ 与 $\mathfrak{RW}_{\mathrm{nonce}}(T_j) = \{(s, k_j)\}$ 满足严格正交不交性 $\mathfrak{RW}_{\mathrm{nonce}}(T_i) \cap \mathfrak{RW}_{\mathrm{nonce}}(T_j) = \varnothing$。在生产级 *x402 Compliance Settlement Gateway* 中，系统默认常驻预置 $K = 32$ 条无锁并发微支付通道，固定占用与串行主通道物理隔离的正交键域 $\mathcal{K}_{32} = \{101, 102, \ldots, 132\}$，通过原子无锁轮询算子 $\ell = (c_{\mathrm{rr}}\text{++}) \bmod 32 \mapsto k_\ell = 101 + \ell$ 将聚合到达率 $\lambda$ 均匀解耦为 32 路速率为 $\lambda/32$ 的独立子流。

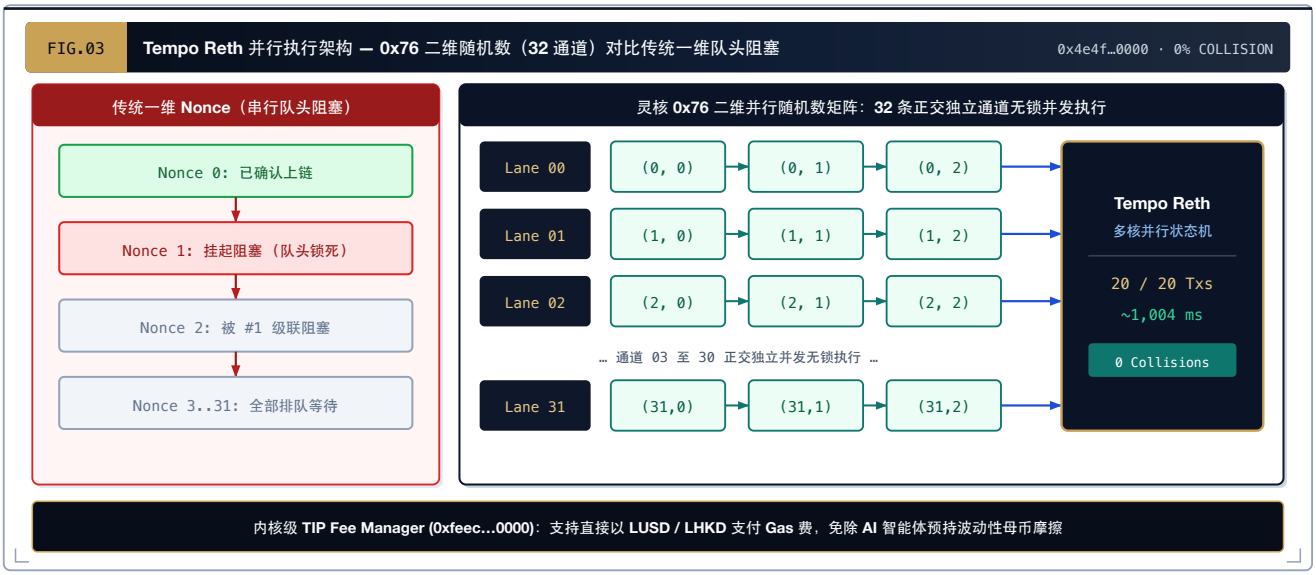



图 **03 — 0x76** 二维并行随机数（**32** 独立通道 × 序列维）无锁并发矩阵与原生稳定币代付 **Gas** 架构

注：实测同一智能体并发派发 20 笔 x402 微支付耗时 ~1,004 ms，实现 100% 吞吐且零 Nonce 碰撞回滚。

### 4.2 0x76 原生交易信封、原子多调用批处理与 Passkey 密钥链授权

传统基于 ERC-4337 的外挂式账户抽象依赖链下打包器（Bundler）内存池模拟与高昂的 EntryPoint 合约重入检查，额外引入数万至数十万 Gas 的固定解释损耗。Linghe Chain 在 EIP-2718 交易类型多路复用层原生定义了类型标识符为 0x76 的协议级事务信封 TxEnvelopeTempo，将批量调用、二维并发坐标、协议级时间锁、稳定币代付以及硬件通行密钥（Passkey / WebAuthn）授权直接下沉为共识状态机的第一公民：

$$\mathrm{Tx}_{0x76} = \texttt{0x76} \,\|\, \mathrm{RLP}([\mathrm{chainId},\ \hat{f}_{\mathrm{prio}},\ \hat{f}_{\max},\ G_{\mathrm{limit}},\ \mathrm{calls},\ \mathcal{L}_{\mathrm{acc}},\ k,\ \nu,\ t_{\mathrm{before}},\ t_{\mathrm{after}},\ t_{\mathrm{fee}},\ \sigma_{\mathrm{payer}},\ \mathcal{L}_{\mathrm{auth}},\ \mathcal{K}^{?}_{\mathrm{auth}},\ \sigma_{\mathrm{sender}}])$$

表 4-1 严格界定了 0x76 信封载荷各 RLP 序列槽的二进制结构、位宽域约束与协议层状态机语义：

| RLP 槽序 | 符号标识 | 协议字段名称与二进制类型 | 协议内核校验规则与执行语义 |
|---|---|---|---|
| 0 | 0x76（前缀） | TxType（uint8，非 RLP 包内） | EIP-2718 交易类型多路复用前缀字节，路由至 Tempo 原生二维并行校验管线 |
| 1 | chainId | chain_id（uint64） | 强制等于 88888（0x15b38），从代数层面阻断跨链重放攻击 |
| 2–4 | $\hat{f}_{\mathrm{prio}}, \hat{f}_{\max}, G_{\mathrm{limit}}$ | max_priority_fee / max_fee / gas_limit | 遵循 EIP-1559 动态费率上限；有效单价 $f_{\mathrm{eff}} = \min(\hat{f}_{\max}, f_{\mathrm{base}} + \hat{f}_{\mathrm{prio}})$ |
| 5 | calls | calls（RLPList<CallTuple>） | 原子批处理向量 $[(c_j, v_j, d_j)]_{j=1}^{m}$（目标地址 20B、主币值 uint256、调用载荷 bytes） |
| 6 | $\mathcal{L}_{\mathrm{acc}}$ | access_list（EIP-2930 List） | 预声明跨合约热状态槽集合，供 Tempo Reth 执行器提前执行异步内存预取 |
| 7–8 | $(k, \nu)$ | nonce_key（uint192）/ nonce（uint64） | 路由至 0x4e4f4e4345000000000000000000000000000000 校验 $\mathcal{N}(s,k) = = \nu$ 并原子自增 |
| 9–10 | $t_{\mathrm{before}}, t_{\mathrm{after}}$ | valid_before / valid_after（uint64?） | 协议层时间窗硬门禁：强制断言 $t_{\mathrm{after}} < t_{\mathrm{block}} < t_{\mathrm{before}}$，超时交易零 Gas 丢弃 |
| 11–12 | $t_{\mathrm{fee}}, \sigma_{\mathrm{payer}}$ | fee_token（address?）/ fee_payer_sig | 指定 6 位精度结算稳定币地址及可选的 0x78 域分离第三方 Gas 赞助机构签名 |
| 13–14 | $\mathcal{L}_{\mathrm{auth}}, \mathcal{K}^{?}_{\mathrm{auth}}$ | authorization_list / key_authorization | EIP-7702 代码委托列表与 0xaaaaaaaa0000000000000000000000000000000000000000 密钥链/Passkey 授权凭证 |
| 15 | $\sigma_{\mathrm{sender}}$ | sender_signature（SignatureEnvelope） | 支持标准 secp256k1、硬件 P-256 Passkey 或经密钥链授权的受限会话子密钥签名 |

该协议层信封实现了三项核心系统能力：

1. 单事务原子多调用批处理（**Atomic Multi-Call Batching**）：调用向量 $\mathrm{calls} = [(c_1, v_1, d_1), \ldots, (c_m, v_m, d_m)]$ 在同一 EVM 事务检查点（Checkpoint）下顺序执行。在自治智能体微支付场景中，协调器无需先后广播“稳定币扣款”与“链上履约评分”两笔独立交易，而是将原生 TIP-20.transfer(payTo, amount) 与 LingheERC8004ReputationRegistry

（`0x8004000000000000000000000000000000000002`）的 `submitVerifiedFeedback` 封装于同一笔 `0x76` 信封（按线性预算模型 $G_{\text{limit}} = 1,500,000 + 900,000 \times m$ 分配算力上限）。一旦任意子调用 $j \in \{1, \dots, m\}$ 返回回滚标志，内核立即回退整组 calls 状态变更，从根源上消除“资金已划转但信誉凭证未入账”的半完成不一致态。

2. 原生密钥链预编译与无迁移 **Passkey** 智能账户（`Account Keychain` `0xaaaaaaaa0000000000000000000000000000000000`）：`0x76` 信封中的可选授权槽 $\mathcal{K}^{?}_{\text{auth}}$ 直接对接内核预编译 `Account Keychain`（`0xaaaaaaaa0000000000000000000000000000000000`），允许主账户在不暴露冷端助记词的前提下，向链上注册具备到期时间与额度约束的操作子密钥或 WebAuthn / Passkey 硬件公钥。与此同时，结合 `EIP-7702` 委托列表 $\mathcal{L}_{\text{auth}}$ 与零迁移智能账户合约 `LingheSmartAccount7702`（以及有界会话密钥合约 `LingheSessionKeyManager` `0x8402000000000000000000000000000000000001`），任何标准 `BIP-39` 外部账户均可原地挂载硬件安全隔区（Apple Secure Enclave Touch ID / Face ID）`P-256` 公钥 $(Q_x, Q_y)$（键标识 keyId = keccak256("LINGHE_P256_PASSKEY_V5" ‖ $Q_x$ ‖ $Q_y$)），直接通过 `ERC-7579` 单调用模式（`0x00`）或批量调用模式（`0x01`，选择器 `0xe9ae5c53`）完成免助记词硬件级交易签署。
3. 原生多稳定币 **Gas** 赞助与异构验证者自动清分（`TIP Fee Manager` `0xfeec000000000000000000000000000000000000`）：为消除智能体或企业客户必须预先采购波动性母币才能发起首笔交易的财务摩擦，`0x76` 交易支持直接指定 `LUSD`（`0x20C0000000000000000000000000000000000001`）或 `LHKD`（`0x20C0000000000000000000000000000000000002`）作为 `feeToken`，或启用第三方机构 Gas 代付（Fee Sponsorship）。在代付模式下，发送方签署自身业务意图，赞助网关则对显式绑定发送方地址 `sender` 的 `0x78` 域分离载荷生成 `secp256k1` 代付签名 $\sigma_{\text{payer}}$：

$$H_{\text{feePayer}} = \text{keccak256}(\texttt{0x78} \,\|\, \text{RLP}([\text{chainId}, \hat{f}_{\text{prio}}, \hat{f}_{\max}, G_{\text{limit}}, \text{calls}, \mathcal{L}_{\text{acc}}, k, \nu, t_{\text{before}}, t_{\text{after}}, t_{\text{fee}}, \text{sender}, \mathcal{L}_{\text{auth}}, \mathcal{K}^{?}_{\text{auth}}]))$$

在区块执行阶段，预编译 `TIP Fee Manager`（`0xfeec000000000000000000000000000000000000`）对实际付费主体 $p \in \{\text{sender}, \text{feePayer}\}$ 持有的支付币种 $t_u \in \{\text{LHC}, \text{LUSD}, \text{LHKD}\}$ 与当值出块验证者 $\upsilon_b$ 注册的目标结算币种 $t_\upsilon = \text{validatorTokens}[\upsilon_b]$ 执行五步原子清算：（i）执行前按 6 位小数精度预扣最大潜在费用上限 $F_{\max} = G_{\text{limit}} \cdot f_{\text{eff}}$，若余额不足（`InsufficientFeeTokenBalance`）则在进入 EVM 前直接拒绝，杜绝算力耗尽攻击；（ii）执行 calls 向量计量真实算力 $G_{\text{used}} \le G_{\text{limit}}$；（iii）原路即时退还未使用差额 $\Delta F_{\text{refund}} = (G_{\text{limit}} - G_{\text{used}}) \cdot f_{\text{eff}}$，实际净扣款 $F_{\text{actual}} = G_{\text{used}} \cdot f_{\text{eff}}$；（iv）若 $t_u = t_\upsilon$（如香港节点出块槽直接收取 `LHKD`），资金零滑点直通入账；若 $t_u \neq t_\upsilon$（如智能体以 `LUSD` 支付 Gas 而当值香港节点要求结算 `LHKD`），内核在同一原子事务内调用 `0xfeec000000000000000000000000000000000000` 内置恒定规则 `Fee AMM` 储备池 $(R_{t_u}, R_{t_\upsilon})$ 完成自动兑付 $\Phi_{\text{out}}(F_{\text{actual}})$；（v）累加至 `collectedFees[v_b][t_v]` 并强制施加块内防操纵锁（`CannotChangeWithinBlock`），严禁提议者在同一区块中途篡改 `validatorTokens` 实施三明治套利。

### 4.3 多通道阿贝尔交换性证明与高并发突发结算实证

**定理 4.1 — 32 通道并发结算可交换性与零碰撞定理**

设清算主账户 $s \in \mathcal{A}$ 在时刻 $t_0$ 的 `TIP-20` 稳定币初始原子余额为 $B_0(s) \in \mathbb{N}_0$。考虑在同一区块窗口或连续流水线槽内并发注入的 $m$ 笔（$1 \le m \le 32$）`0x76` 批量结算交易集合 $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$，其中每笔交易 $T_i$ 分配至两两互异的正交二维随机数通道键 $k_i \in \mathcal{K}_{32}$（即 $\forall i \neq j,\ k_i \neq k_j$），携带当前通道有效序号 $\nu_{s,k_i} = \mathcal{N}(s, k_i)$、最大费用锁定预算 $F_{\max,i}$、向目标收款方 $r_i \neq s$ 划转的合规稳定币金额 $a_i > 0$，以及向 `LingheERC8004ReputationRegistry`（`0x8004000000000000000000000000000000000002`）写入的唯一任务哈希槽 $u_i$（$\forall i \neq j,\ u_i \neq u_j$）。若充裕流动性前提

$$B_0(s) \ge \sum_{i=1}^{m} (a_i + F_{\max,i})$$

成立且所有参与主体均通过 `TIP-403` 授权检查，则：（1）零随机数碰撞性：$\mathcal{T}$ 中任意两笔交易在交易池准入与共识排序阶段发生 Nonce 抢占冲突的概率恒为 $P_{\text{collision}} \equiv 0$；（2）严格阿贝尔状态交换律：对 $m$ 阶对称群 $S_m$ 中的任意区块内排列 $\pi \in S_m$，复合状态转换算子收敛至完全相同的唯一终态账本：

$$(\Upsilon_{T_{\pi(m)}} \circ \Upsilon_{T_{\pi(m-1)}} \circ \cdots \circ \Upsilon_{T_{\pi(1)}})(\Sigma_0) = \Sigma_{\text{post}}, \qquad \forall \pi \in S_m$$

**证明**

将全局状态空间 $\Sigma$ 按受 $\mathcal{T}$ 影响的四类存储子流形做直和分解 $\Sigma_{\mathcal{T}} = \Sigma_{\text{nonce}} \oplus \Sigma_{\text{rep}} \oplus \Sigma_{\text{bal}} \oplus \Sigma_{\text{fee}}$，分别检验任意对换算子 $(i, j) \in S_m$ 下的不交性与结合交换律：

1. 二维随机数子流形 $\Sigma_{\text{nonce}}$ 正交性与零碰撞：在 `2D Nonce Manager`（`0x4e4f4e4345000000000000000000000000000000`）中，交易 $T_i$ 仅读取并更新键槽 $(s, k_i)$。由前提 $\forall i \neq j,\ k_i \neq k_j$，各交易在 $\Sigma_{\text{nonce}}$ 上的读写集两两互不相交 $\{(s,k_i)\} \cap \{(s,k_j)\} = \varnothing$。因此交易池验证器对 $T_i$ 检验 $\mathcal{N}_0(s,k_i) = = \nu_{s,k_i}$ 的真值完全独立于其余 $m-1$ 笔交易的到达次序或打包状态，Nonce 碰撞概率恒满足 $P_{\text{collision}} \equiv 0$；对任意排列 $\pi \in S_m$，执行后各通道序列号唯一确定为 $\mathcal{N}_{\text{post}}(s,k_i) = \nu_{s,k_i} + 1$。
2. 信誉存证子流形 $\Sigma_{\text{rep}}$ 写集互斥性：在 `0x8004000000000000000000000000000000000002` 中，每笔交易依据互异的任务或支付签名摘要 $u_i$ 写入独立映射槽 `recordedTaskHashes[u_i]` 并追加评价记录，键空间交集为空；声誉总分累加器 $\sum_{i=1}^{m} \text{score}_i$ 与总次数计数器 $m$ 在 `uint256` 无溢出域上严格满足整数加法交换律。
3. 稳定币余额子流形 $\Sigma_{\text{bal}}$ 与费用子流形 $\Sigma_{\text{fee}}$ 阿贝尔半群性质：任取排列 $\pi \in S_m$ 及任意执行前缀步 $r \in \{1,\dots,m\}$。在第 $r$ 步前置扣锁瞬间，账户 $s$ 的瞬时可用余额下界满足：

$$B^{(r-1)}(s) - (a_{\pi(r)} + F_{\max,\pi(r)}) \geq B_0(s) - \sum_{j=1}^{r} (a_{\pi(j)} + F_{\max,\pi(j)}) \geq B_0(s) - \sum_{i=1}^{m} (a_i + F_{\max,i}) \geq 0$$

故在任意排列路径上，底层 Rust `uint256` 无符号减法绝不触发算术下溢回滚（Underflow Revert），每笔交易的原子扣款与未用 Gas 退税 $F_{\text{actual},\pi(r)} = G_{\text{used},\pi(r)} \cdot f_{\text{eff},\pi(r)}$ 均确定性成功。由于 $(\mathbb{Z},+)$ 上的无下溢加减运算构成阿贝尔群，主账户终态余额恒收敛于与排列 $\pi$ 无关的不变量 $B_{\text{post}}(s) = B_0(s) - \sum_{i=1}^{m} (a_i + F_{\text{actual},i})$，任意收款账户 $r^*$ 的终态余额亦唯一收敛于 $B_{\text{post}}(r^*) = B_0(r^*) + \sum_{i:\, r_i = r^*} a_i$。即使在恶意攻击者故意构造超额并发流破坏前提 $B_0(s) < \sum (a_i + F_{\max,i})$ 的退化边界下，超出可用余额的后序通道交易在内核层独立触发原子回滚，已成交前缀账本与全局货币总供应量守恒丝毫不受污染。

在 `ChainID: 88888` 全栈生产压力基准实测中，测试集群向 32 通道并发池同时突发注入 `20` 笔自治智能体 `0x76` 原子批处理微支付请求（映射至正交通道键 $k \in \{101,\dots,120\}$，总计包含 `40` 次内部合约状态写入、前置 `TIP-403` 筛查与硬件飞地回执签名），全批次 `20/20` 笔并发结算 100% 终局确认总耗时仅 `1,004 ms`，链上与网关层观测到的 Nonce 碰撞与重排次数严格等于 `0`。证毕。

Q.E.D. ■

### 4.4 稳定币代付五阶段价值守恒与并发对抗边界分析

为确保 `0x76` 交易信封在恶意网络重放、并发超支竞态与三明治费率操纵下保持严格确定性，`Tempo Reth` 与 `TIP Fee Manager`（`0xfeec000000000000000000000000000000000000`）在底层联合构筑了表 4-2 所示的五阶段状态机闭环与两项核心对抗防线：

| 执行阶段 | 状态机操作原语 | 精确账本与储备池状态转换方程 | 异常拒绝门禁（**Fail-Closed**） |
|---|---|---|---|
| **I.** 预扣锁 | `lockMaxFeeBudget` | $B'_{t_u}(p) = B_{t_u}(p) - G_{\text{limit}} \cdot f_{\text{eff}}$ | 余额不足立即拒绝 `InsufficientFeeTokenBalance` |
| **II.** 批执行 | `dispatchCallsVector` | 顺序迭代 $[(c_j, v_j, d_j)]_{j=1}^{m}$，累计 $G_{\text{used}} \leq G_{\text{limit}}$ | 任一子调用失败则整组 `calls` 检查点原子回滚 |
| **III.** 差额退税 | `refundUnusedGas` | $B_{\text{post},t_u}(p) = B'_{t_u}(p) + (G_{\text{limit}} - G_{\text{used}}) \cdot f_{\text{eff}}$ | 严格保证 $B_{\text{pre},t_u}(p) - B_{\text{post},t_u}(p) = F_{\text{actual}}$ |
| **IV.** 异构兑付 | `settleFeeAmmOrDirect` | 若 $t_u \neq t_v$：$R'_{t_u} = R_{t_u} + F_{\text{actual}}$，$R'_{t_v} = R_{t_v} - \Phi_{\text{out}}(F_{\text{actual}})$ | 池深度不足回退，杜绝悬空无准备金记账 |
| **V.** 验证者入账 | `accrueValidatorFee` | $\text{collectedFees}'[v_b][t_v] = \text{collectedFees}[v_b][t_v] + \Phi_{\text{out}}(F_{\text{actual}})$ | 同块算改计价币种触发 `CannotChangeWithinBlock` |

1. 跨层重放与滞留过期双重截断：每笔 `0x76` 载荷同时受 `chainId = 88888`、通道单调计数器 $\nu_{s,k}$ 以及时间窗 $(t_{\text{after}}, t_{\text{before}})$ 三重正交约束。一旦网络分区恢复后旧签名包迟到且满足 $t_{\text{block}} \geq t_{\text{before}}$，共识验证入口直接以过期无效丢弃，不消耗任何链上用户余额。

2. 受制裁主体借道干净代付者逃逸阻断：若被列入 `TIP-403 Policy #2` 黑名单的地址试图借助合规企业 `feePayer` 的 `0x78` 代付签名转移受规管资产，尽管阶段 I 费用扣自合规代付者，但在阶段 II 执行资产划转瞬间，Rust 内核仍对实际资产发送方与接收方执行强制筛查并原子回滚内部调用。

---

第 05 章 · SECTION 05

# 内核级金融与监管状态机原语（TIP-20 · TIP-403 · 原生限价订单簿 DEX）

通用公有链将法币稳定币、反洗钱制裁拦截与外汇流动性池全部堆叠于用户态智能合约层，由此引发两项根本性系统缺陷：其一，外挂合约黑名单在多跳组合调用与包装资产（Wrapped Tokens）面前极易被穿透失效；其二，恒定乘积自动做市商（AMM，$x \cdot y = k$）在面值附近流动性利用率极低，大额机构换汇承受严重非线性滑点。`Linghe Chain` 在 `Tempo Reth` Rust 执行内核层直接固化了 `TIP-20` 原生协议资产、`TIP-403 Policy Registry` 合规策略引擎与 `Enshrined StablecoinDEX` 中央限价订单簿三大金融原语，并与创世全链结算器 `LingheOmnichainSettler`（`0x8900000000000000000000000000000000000001`）的香港联系汇率走廊严密咬合。

## 5.1 创世内生 `TIP-20` 双轨账本原语体系与 `TIP-403` 内核策略耦合

由协议与软件系统研发主体 `MYRILUM LIMITED`（迈瑞伦有限公司，香港特别行政区《公司条例》第 **622** 章私人股份有限公司，公司注册编号 / 唯一商业识别码 **UBI**：`80912065`，商业登记证号：`80912065-000-07-26-0`）独立设计实现的灵核协议状态机，在创世第 `0` 块原生固化了三项统一采用 `6` 位小数精度（1 单位 $= 10^6$ 原子单位）的 `TIP-20` 协议内生标准账本原语与非流通确定性基准测试状态，严格执行共识安全计量原语与定值清算参考原语的双轨状态空间正交隔离（**Orthogonal State-Space Separation**）：

| 协议账本原语 | 创世预部署精确地址（`TIP-20`） | 精度 | 状态机职能定位与确定性基准测试状态锚定 | 内核策略绑定状态 |
|---|---|---|---|---|
| `LHC` | `0x20C0000000000000000000000000000000000000` | `6` | 验证者共识质押权重计量、系统安全预算与惩罚吸收账本原语（非流通基准状态） | `Policy #1`（`OPEN`） |
| `LUSD` | `0x20C0000000000000000000000000000000000001` | `6` | 1:1 美元 HQLA 参考模型账本原语；确定性基准负债状态 `$10,000,000.00`，参考储备状态 `$10,050,000.00`（`100.50%`） | 跨链与联汇海关协同筛查 |
| `LHKD` | `0x20C0000000000000000000000000000000000002` | `6` | 形式化建模香港《稳定币条例》（`Cap. 656`）不变量之 1:1 港元参考账本原语；基准负债状态 `HK$78,000,000.00`，参考储备状态 `HK$78,936,000.00`（`101.20%`） | 内核强绑定 `Policy #2`（`ENFORCED`） |

在形式化状态机与协议参考偿付模型中，`LUSD` 与 `LHKD` 的链上基准负债状态 $L_t^{(i)}$ 在合约字节码层强制受独立破产隔离高流动性优质资产（即期现金存款与剩余期限 $\leq 90$ 天政府外汇基金票据/短期国库券）参考储备状态 $R_t^{(i)} \geq L_t^{(i)}$ 约束（确定性基准测试状态锚定于 `LingheComplianceReserveVault` `0x8800000000000000000000000000000000000001`：其中 `LUSD` 基准储备默克尔根为 `0x5c3f82332fbffc0be49434232a9a97e2bf2c797423aa3fde4afd19b2c264f410`，`LHKD` 基准储备默克尔根为 `0xa3e5fd1ee51db26128c49f848bc4eb6d71443fbfee51a40b9159832a21beae6e`）。由于协议内核严禁 `LHC` 与任一定值清算参考原语之间发生算法铸销互换，参考偿付覆盖率 $\rho_t^{(i)} = R_t^{(i)} / L_t^{(i)}$ 对 `LHC` 计价参数 $P_{\mathrm{LHC}}(t)$ 的一阶偏导恒满足 $\partial \rho_t^{(i)} / \partial P_{\mathrm{LHC}}(t) \equiv 0$，从数

学根源上消除了内生抵押自反性螺旋。同时，`LHKD` 与 `LUSD` 在字节码层严格锁定零生息率不变量（持有人生息率 $r_{\text{holder}} \equiv 0$，不含任何重定基或收益分配代码），在计算机科学层面精确刻画香港《稳定币条例》（`Cap. 656`）纯支付清算原语与非集体投资计划（`Cap. 571`）的状态约束。

为实现零旁路反洗钱与反恐融资（AML/CFT）实时冻结，每个 `TIP-20` 资产在 `Tempo Reth` Rust 状态结构中内嵌 64 位无符号策略指针槽 `transferPolicyId`，直连内核预编译合约 `TIP-403 Policy Registry`（`0x403c0000000000000000000000000000000000`）。合规管理员通过 `createPolicyWithAccounts(admin, policyType = 1, blockedAccounts)` 在内核注册表中实例化黑名单策略 `Policy #2`（实时同步联合国安理会 UNSC 综合制裁名单、OFAC SDN 名单及香港宪报第 537 章与第 575 章指定实体），并调用 `LHKD` 的 `changeTransferPolicyId(2)` 完成底层绑定。此后，每当执行引擎触发任何涉及该 `TIP-20` 资产余额变动的状态原语（`transfer`、`transferFrom`、批量代扣或原生撮合交割），Rust 内核在改写账户余额映射前直接于进程内内存表执行 $O(1)$ 查表判定：

$$\text{Auth}_{\text{TIP-403}}(\text{token}, \text{from}, \text{to}) \equiv (\pi_{\text{pol}}(\text{token}) \leq 1) \ \vee \ (\texttt{isAuthorized}(\pi_{\text{pol}}(\text{token}), \text{from}) \ \wedge \ \texttt{isAuthorized}(\pi_{\text{pol}}(\text{token}), \text{to}))$$

### 5.2 内核原生中央限价订单簿 `Enshrined StablecoinDEX`（`0xdec0000000000000000000000000000000000000`）撮合代数

针对二级市场高频稳定币做市与跨链求解器库存再平衡，`Linghe Chain` 在地址 `0xdec0000000000000000000000000000000000000` 处固化了 Rust 内核级中央限价订单簿预编译 `Enshrined StablecoinDEX`。该引擎摒弃连续曲线自动做市商，围绕报价基准资产 `LHC`（`0x20C0000000000000000000000000000000000000`）为 `LUSD/LHC` 与 `LHKD/LHC` 建立基于双向链表的离散刻度订单簿（CLOB）。

设标度基数常量为 $\text{PRICE_SCALE} = 100,000$，离散整数刻度为 $\tau \in \mathbb{Z}$。内核强制锁定合法刻度域 $\mathcal{T}_{\text{tick}}$ 与精确线性定价映射 $P(\tau)$ 为：

$$P(\tau) = \frac{\text{PRICE_SCALE} + \tau}{\text{PRICE_SCALE}} = 1 + \frac{\tau}{100,000}, \qquad \tau \in [-2000, \ +2000] \ \wedge \ \tau \equiv 0 \ (\text{mod } 10)$$

其中，最小刻度步长 $\Delta\tau = \text{TICK_SPACING} = 10$ 精确对应 1 bps（0.01%）的机构外汇报价粒度，允许挂单价格走廊严格收窄于面值附近的 $[0.98000, \ 1.02000]$（即 `1.00000` ±2.00%），单笔最小挂单门槛设为 $\text{MIN_ORDER_AMOUNT} = 100,000,000$（`100.000000` 单位，阻断粉尘挂单耗尽链表内存攻击）。订单簿内部严格执行价格-时间优先（**Price-Time Priority / Strict FIFO**）确定性撮合律：

1. 最优价格寻址与 **FIFO** 队列消耗：买方挂单簿（Bid Book）按刻度严格降序 $\tau_{\text{bid}}^{(1)} > \tau_{\text{bid}}^{(2)} > \dots$ 组织，卖方挂单簿（Ask Book）按刻度严格升序 $\tau_{\text{ask}}^{(1)} < \tau_{\text{ask}}^{(2)} < \dots$ 组织；同一刻度 $\tau$ 下的挂单节点按上链序列号 $\omega_1 < \omega_2$ 维护双向链表队列 $Q_\tau = [O_{\tau,\omega_1}, O_{\tau,\omega_2}, \dots]$。当吃单者调用 `swapExactAmountIn(tokenIn, tokenOut, amountIn, minAmountOut)` 时，Rust 撮合算子从对手盘最优刻度 $\tau^*$ 队首开始逐单扣减剩余量 $q_{\tau^*,\omega_j}$。只要最优刻度累计深度 $\sum_j q_{\tau^*,\omega_j} \geq \text{amountIn}$，整笔大额交易即以单一固定刻度价格 $P(\tau^*)$ 零曲线边际滑点完成交割；若跨越多个刻度后总输出低于 `minAmountOut`，整笔交易原子回滚。
2. 原生自动翻转做市单（**Auto-Flipping Liquidity Orders**，`placeFlip`）：做市商可通过 `placeFlip(token, amount, isBid, tick, flipTick)` 注入自平衡流动性。设买单部署于 $\tau_{\text{bid}} = -10$（买入价 $P(-10) = 0.99990$），翻转目标刻度设为 $\tau_{\text{flip}} = +10$（卖出价 $P(+10) = 1.00010$）。一旦该买单在队列中被吃单流 100% 消耗，Rust 状态机在同一指令周期内自动将所得头寸反转为挂在 $\tau_{\text{flip}} = +10$ 队尾的等规模卖单，无需任何外部预言机喂价或链下撤挂单交易即可常驻维持 `2 bps`（`0.02%`）的极窄双边点差。在生产实测中，向 `0xdec0000000000000000000000000000000000000` 提交 `100.000000 LUSD` 市价吃单，在单一 `250ms` 槽内按刻度 `-10` 零额外滑点精确交割 `99.990000 LHC`（`99,990,000` 原子单位）。

### 5.3 香港金管局联系汇率走廊（`7.8000`）算术与内核不可绕过性定理

在美元稳定币 `LUSD` 与港元稳定币 `LHKD` 的直接法币交叉换汇层，创世合约 `LingheOmnichainSettler`（`0x8900000000000000000000000000000000000001`）将香港金融管理局（HKMA）联系汇率制度（Linked Exchange Rate System）自 2005 年确立的强方兑换保证（`7.7500 HKD/USD`）与弱方兑换保证（`7.8500 HKD/USD`）直接编码为不可逾越的智能合约宪法边界。

合约以标度因子 $S_{\text{FX}} = 10,000$ 定义无符号 32 位汇率状态变量 `hkmaPegRateScaled`（记为 $\pi_{\text{HKMA}}$，默认创世中枢值 $\pi_0 = 78000 \Leftrightarrow 7.8000$ HKD/USD）。治理函数 `setBorderPolicy(uint32 newPegRateScaled, bool paused)` 同时实施主权门限锁与走廊区间硬断言：第一，`msg.sender` 必须严格等于香港、新加坡、东京、苏黎世四法域 `3-of-4` 主权门限多

签合约 `LingheSovereignTimelockMultisig`（`0x8900000000000000000000000000000000000004`），且在 `v5.0` 架构下受控于 `LingheSovereignSecurityGuardV5` 的强制时间锁 eta = $t_{\text{queue}} + \Delta_{\min}$；第二，即便获得跨法域一致表决，字节码仍强制核验 `77500 <= newPegRateScaled && newPegRateScaled <= 78500`，任何越出闭区间 $[7.7500, 7.8500]$ 的参数变更必然触发 `OUTSIDE_HKMA_BAND` 确定性回滚。

对于通过双端 `TIP-403` 合规校验的兑换调用 `swapLinkedRate(tokenIn, amountIn, recipient)`，合约执行如下双向向零整除算术：

$$A^{\text{out}}_{\text{LHKD}} = \left\lfloor \frac{A^{\text{in}}_{\text{LUSD}} \cdot \pi_{\text{HKMA}}}{10,000} \right\rfloor, \qquad A^{\text{out}}_{\text{LUSD}} = \left\lfloor \frac{A^{\text{in}}_{\text{LHKD}} \cdot 10,000}{\pi_{\text{HKMA}}} \right\rfloor$$

定义联汇池标度总净值泛函为 $W(B_{\text{USD}}, B_{\text{HKD}}) = B_{\text{USD}} \cdot \pi_{\text{HKMA}} + 10,000 \cdot B_{\text{HKD}}$。对任意正整数输入 $x, y \in \mathbb{Z}_{>0}$，正向兑换净值增量为 $\Delta W_{+} = x\pi_{\text{HKMA}} - 10,000\lfloor x\pi_{\text{HKMA}}/10,000 \rfloor = (x\pi_{\text{HKMA}} \bmod 10,000) \geq 0$，逆向兑换净值增量为 $\Delta W_{-} = 10,000y - \pi_{\text{HKMA}}\lfloor 10,000y/\pi_{\text{HKMA}} \rfloor = (10,000y \bmod \pi_{\text{HKMA}}) \geq 0$。整数截断余数恒定留存池内且复合往返映射严格收缩，从算术根源上免疫高频粉尘往返抽干攻击（Rounding-Drain Exploit）。进一步地，设一级市场法币申赎摩擦率为 $c_{\text{mint}}, c_{\text{redeem}} \geq 0$，赎回交割时滞为 $\Delta t$，无风险拆借利率为 $r_f$。依托香港转数快（HKICL FPS，`T+0` 实时交割）与港元即时支付结算系统（HKD CHATS / RTGS，日内 `T+0` 交割）将 $\Delta t \to 0$ 压缩归零，配合二级市场 `0xdec0000000000000000000000000000000000000` 的 `2 bps` 极窄限价深度与 `250ms` 零重组终局，二级市场隐含法币价格 $p_t^{\text{mkt}}$ 被跨市场无套利机制刚性夹逼于极窄平价走廊：

$$1 - c_{\text{redeem}} - r_f \cdot \Delta t \leq p_t^{\text{mkt}} \leq 1 + c_{\text{mint}} + r_f \cdot \Delta t_{\text{mint}} \quad \xrightarrow{\Delta t \to 0} \quad p_t^{\text{mkt}} \in [1 - c_{\text{redeem}},\, 1 + c_{\text{mint}}]$$

表 5-2 汇总了内核级 `TIP-403` 与创世系统合约在全部五维价值入口相对于传统外挂合约黑名单的完备闭合优势：

| 复杂组合调用或跨境攻击路径 | 传统外挂 **ERC-20** 黑名单表现 | 灵核内核与系统合约协同拦截点及确定性状态结果 |
|---|---|---|
| 直连转账、`0x76` 批处理或代付 **Gas** 转账 | 易被未设防中继合约绕过 | `Tempo Reth` 内核 `0xef` 短路强制查验 `0x403c0000000000000000000000000000000000000000` 并硬回退 |
| 包装合约（**Wrapper**）与嵌套 `DELEGATECALL` | 份额代币二级流转完全逃逸冻结 | 真实账本驻留 `0x20C0000000000000000000000000000000000002` 内核空间，底层解包即刻拒绝 |
| **CLOB** 限价簿挂单与吃单（`placeFlip` / `swap`） | 外部 AMM LP 凭证可二次流转 | `0xdec0000000000000000000000000000000000000` 共享同一 Rust `TIP-20` 转移钩子拦截 |
| 受限源链地址借干净 **Solver** 跨链入境（`ERC-7683`） | 穿透失效（仅见干净 Solver） | `0x8900000000000000000000000000000000000001` 双端校验回滚 `OMNICHAIN_ORIGIN_SANCTIONED` |
| 持有效 **Groth16** 证明请求跨链桥释放或联汇换汇 | 密码学桥无视名单直接放款 | `swapLinkedRate` 回滚 `FX_TIP403_SANCTIONED`；零知识桥回滚 `ZK_BRIDGE_TIP403_SANCTIONED` |

**不变量 5.1 — 内核级制裁策略不可绕过性（Kernel Non-Bypassability）**

设区块高度 $h$ 时 `TIP-403 Policy Registry`（`0x403c0000000000000000000000000000000000000000`）在策略 `Policy #2` 下的受限地址集合为 $\mathcal{S}_h = \{a \in \mathcal{A} \mid \neg\text{isAuthorized}(2, a)\}$。对于绑定该策略的受规管协议资产 `LHKD`（`0x20C0000000000000000000000000000000000002`）及受跨境海关守卫保护的结算通道，不存在任何用户态合约代理、批量信封或跨链证明构造，能够使任一受限主体 $u \in \mathcal{S}_h$ 在有效提交的状态转换中产生非零受规管资产净变动：

$$\forall u \in \mathcal{S}_h, \quad \forall \tau \in \mathcal{T}_{\text{committed}}(h), \qquad B^{\text{LHKD}}_{h+1}(u) - B^{\text{LHKD}}_{h}(u) \equiv 0$$

**定理 5.1 — 任意嵌套调用栈下的原子合规拦截完备性**

设任意链上事务 $\tau$（涵盖标准 `0x02` 交易、`0x76` 多调用原子批处理、`EIP-7702` 委托执行、跨链意图填充与零知识桥释放）在 EVM 执行期间展开的有向有序内部调用树为 $\mathcal{C}(\tau) = (V, E)$，调用栈最大嵌套深度为 $d_{\max} \in [0, 1024]$。若调用树中任意深度 $d \le d_{\max}$ 的叶节点或中间帧试图：（a）变更受限主体 $u \in \mathcal{S}_h$ 的 `LHKD` 余额，或（b）在 `LingheOmnichainSettler`（`0x8900000000000000000000000000000000000001`）与 `LingheZkLightClientBridge`（`0x8900000000000000000000000000000000000002`）中以 $u \in \mathcal{S}_h$ 作为源链委托发起人、换汇交易员或目标链收款人，则该目标帧必以概率 $1$ 触发确定性回滚，且任何未能显式隔离该子帧回滚的祖先事务路径对受规管账本的状态改写恒为零。

**证明**

对调用树 $\mathcal{C}(\tau)$ 中触发受规管价值流转的叶调用帧深度 $d \ge 0$ 及全部协议入口实施分情况完备性归纳：

1. 基例（$d = 0$，顶层直接调用与 `0x76` 批量直调）：
   - 若顶层直接调用或通过第三方 `feePayer` 赞助的 `0x76` `calls` 帧请求 `0x20C0000000000000000000000000000000000002` 转移资金，由于该地址植入 `0xef` 哨兵字节码，执行流绕过一切用户态代理直接进入 `Tempo Reth` Rust `TIP-20` 内核转移算子。该算子在修改存储映射前无条件读取 `transferPolicyId == 2` 并核验 `0x403c0000000000000000000000000000000000`；因 $u \in \mathcal{S}_h \implies \text{isAuthorized}(2, u) = \text{false}$，Rust 内核立即返回硬失败标志，回退该帧及所属 `0x76` 原子批处理包。
   - 若顶层调用内核订单簿 `0xdec0000000000000000000000000000000000000` 的 `place`、`placeFlip` 或 `swapExactAmountIn`，CLOB 锁仓与交割在底层直接调用同一 Rust `TIP-20` 账本变更钩子，同样触发原子拒绝。

2. 归纳步（$1 \le d \le 1024$，任意嵌套包装合约、`DELEGATECALL` 与会话代理）：假设敌手部署任意多层跳板合约栈 $M_1 \xrightarrow{\text{CALL}} M_2 \xrightarrow{\text{DELEGATECALL}} \cdots \xrightarrow{\text{CALL}} M_d$（或经由会话管理器 `0x8402000000000000000000000000000000000001` 与 `Canonical Permit2` `0x000000000022d473030f116ddee9f6b43ac78ba3` 嵌套路由）。由于 `LHKD` 余额槽唯一驻留于受内核保护的 `0x20C0000000000000000000000000000000000002` 地址命名空间下，任何外部合约的 `DELEGATECALL` 仅能改写调用者自身存储槽而绝无权限触碰 `LHKD` 真实账本；欲使 $B^{\text{LHKD}}(u)$ 发生非零改变，调用树叶子末端必在深度 $d+1$ 向 `0x20C0000000000000000000000000000000000002` 发起跨上下文 `CALL`。此时由基例结论，内核预编译对实际资金扣减方与入账方 $u$ 即刻判定违例并销毁第 $d+1$ 帧的全部状态增量。即便外层恶意合约 $M_d$ 使用低级 `call` 捕获布尔返回值 `false` 而不向上冒泡异常，第 $d+1$ 帧自身的账本变更已在子上下文退出瞬间被 EVM 日志回滚清空，故 $B^{\text{LHKD}}_{h+1}(u) - B^{\text{LHKD}}_{h}(u) \equiv 0$ 依然严格成立。
3. 跨链与跨币种边界闭包（阻断代理遮蔽逃逸）：针对外部源链受限地址 $u_{\text{orig}} \in \mathcal{S}_h$ 试图借干净求解器 $S \notin \mathcal{S}_h$ 入境垫付 `LUSD` 或换汇清洗的攻击，`0x8900000000000000000000000000000000000001` 的 `checkBorderCompliance` 动态读取 `LHKD.transferPolicyId()` 并回调 `0x403c0000000000000000000000000000000000`：在 `openCrossChainOrder` 中拦截受限出境方（`OMNICHAIN_TIP403_SANCTIONED`），在 `fillCrossChainOrder` 中双重拦截源链发起人 `originEvmSender` 与目标收款人 `recipient`（`OMNICHAIN_ORIGIN_SANCTIONED` / `OMNICHAIN_RECIPIENT_SANCTIONED`），在 `swapLinkedRate` 中双向拦截调用者与接收方（`FX_TIP403_SANCTIONED`）；而 `0x8900000000000000000000000000000000000002` 的 `verifyProofAndRelease` 则在验证零知识证明前优先核验 `recipient` 的 `Policy #2` 授权态（`ZK_BRIDGE_TIP403_SANCTIONED`）。至此，任意调用栈深度与跨链入口均构成完备闭合。证毕。

Q.E.D. ■

第 06 章 · SECTION 06

# 双椭圆曲线密码学根基：BN254 Groth16 零知识桥与 EIP-7951 规范化 P-256 硬件信任根

主权金融公链在跨链状态终局核验与机密 AI 算力鉴证两端面临截然不同的代数约束：前者要求具备高效双线性配对映射的友好曲线以实现恒定 $O(1)$ 复杂度的简洁非交互零知识证明（zk-SNARK），后者则必须原生兼容全球半导体芯片安全隔区（NVIDIA H100 Confidential Computing、AWS Nitro Enclaves 及硬件安全模块 HSM）在硅片熔丝中固化的 NIST `secp256r1`（`P-256`）椭圆曲线标准。`Linghe Chain` 构建了 `BN254`（`alt_bn128`）配对曲线与规范化低阶 `P-256` 硬件验证曲线双核驱动的密码学根基。

## 6.1 `BN254` 扩域塔、`LingheZkLightClientBridge` 与 `815,381 gas` 配对核验推导

创世零知识轻客户端桥合约 `LingheZkLightClientBridge`（`0x8900000000000000000000000000000000000002`）采用嵌入度 $k = 12$ 的 Barreto-Naehrig 曲线 `BN254`，在字节码常量区严格锁定 254 位素数基域特征 $q_{\mathrm{BN}}$（`BN254_PRIME_Q`）与素数阶循环子群标量域阶 $r_{\mathrm{BN}}$（`BN254_SCALAR_R`）：

$$q_{\mathrm{BN}} = 21888242871839275222246405745257275088696311157297823662689037894645226208583$$
$$r_{\mathrm{BN}} = 21888242871839275222246405745257275088548364400416034343698204186575808495617$$

因 $q_{\mathrm{BN}} \equiv 3 \pmod 4$，多项式 $u^2 + 1$ 在 $\mathbb{F}_{q_{\mathrm{BN}}}$ 上不可约，由此构造二次扩域 $\mathbb{F}_{q_{\mathrm{BN}}^2} = \mathbb{F}_{q_{\mathrm{BN}}}[u]/(u^2+1)$。左源群 $\mathbb{G}_1$ 定义为短 Weierstrass 曲线 $E(\mathbb{F}_{q_{\mathrm{BN}}}): y^2 = x^3 + 3$ 的 $r_{\mathrm{BN}}$ 阶子群（生成元 $G_1 = (1, 2)$）；右源群 $\mathbb{G}_2$ 经 D-type 六次扭曲（扭曲元 $\xi = 9 + u \in \mathbb{F}_{q_{\mathrm{BN}}^2}$）同构映射至扭曲曲线 $E'(\mathbb{F}_{q_{\mathrm{BN}}^2}): y^2 = x^3 + 3/(9+u)$ 的 $r_{\mathrm{BN}}$ 阶子群；目标群 $\mathbb{G}_T = \mu_{r_{\mathrm{BN}}} \subset \mathbb{F}_{q_{\mathrm{BN}}^{12}}^{\times}$ 承载最优 Ate 双线性配对 $e: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$。遵循 `EIP-197` 大端序一阶多项式规范，所有传入预编译 `0x0000000000000000000000000000000000000008` 的 $\mathbb{F}_{q_{\mathrm{BN}}^2}$ 坐标 $(c_0 + c_1\,u)$ 均经坐标置换算子 $\Pi_{\text{EIP-197}}$ 编码为高次虚部在前、常数实部在后的双字结构 `[c1_imag, c0_real]`。

当跨链证明提交至 `verifyProofAndRelease` 时，`computePublicSignal` 首先将域隔离标签、本链 `chainId`（`88888`）、防重放空值器 `nullifierHash`、源链标识、区块高度 $h_{\mathrm{src}}$、源链状态根 `stateRoot`、结算代币、接收地址及原子金额压缩归约为唯一标量域公开输入 $x \in \mathbb{F}_{r_{\mathrm{BN}}}$：

$$x = \mathrm{uint256}(\mathrm{keccak256}(\texttt{"LINGHE_ZK_BRIDGE_V3"} \,\|\, \mathrm{chainId} \,\|\, \mathrm{nullifierHash} \,\|\, \mathrm{sourceChainId} \,\|\, h_{\mathrm{src}} \,\|\, \mathrm{stateRoot} \,\|\, \mathrm{token} \,\|\, \mathrm{recipient} \,\|\, \mathrm{amount})) \bmod r_{\mathrm{BN}}$$

在源链双向托管合约 `LingheOriginEscrow7683` 的零知识清偿入口中，公开信号采用独立域标签 `"LINGHE_ESCROW_7683_ZK_FILL_V4"` 并额外绑定胜出做市商地址 `solver` 与灵核链状态根，彻底封死内存池抢跑者截取有效证明 $(A, B, C)$ 后篡改提款接收方的中间人窃取路径。随后，`verifyGroth16Proof` 执行严格的基域与标量域规范性检查及三阶段预编译流水线：

1. 基域与标量域边界门禁：强制断言 $\mathbb{G}_1$ 证明点坐标满足 $A_x < q_{\mathrm{BN}} \wedge A_y < q_{\mathrm{BN}}$（否则回滚 `INVALID_G1_A`）及 $C_x < q_{\mathrm{BN}} \wedge C_y < q_{\mathrm{BN}}$（否则回滚 `INVALID_G1_C`），杜绝 $A_x + k \cdot q_{\mathrm{BN}}$ 形式的模溢出延展伪造；公开输入经 `scalar = publicInputScalar % BN254_SCALAR_R` 严格约束于 $< r_{\mathrm{BN}}$，且各预编译底层硬性核验曲线方程归属与 $r_{\mathrm{BN}}$ 素数阶子群成员资格。
2. 公开输入线性组合合成（`0x07 ecMul` → `0x06 ecAdd`）：给定链上验证密钥 $\mathrm{vk} = (\alpha_1, \beta_2, \gamma_2, \delta_2, \mathrm{IC}_0, \mathrm{IC}_1)$（由 `0x8900000000000000000000000000000000000004` 主权多签排他受控更新），合约首先静态调用 `0x0000000000000000000000000000000000000007`（`ecMul`，输入 `96B` `[_vk.ic1[0], _vk.ic1[1], x]`，输出 `64B`）计算标量乘点 $M = [x]\mathrm{IC}_1 \in \mathbb{G}_1$；继而静态调用 `0x0000000000000000000000000000000000000006`（`ecAdd`，输入 `128B` `[_vk.ic0[0], _vk.ic0[1], M_x, M_y]`，输出 `64B`）合成公开信号承诺点：
$$\mathrm{vk}_x = \mathrm{IC}_0 + x \cdot \mathrm{IC}_1 \in \mathbb{G}_1$$
3. $\mathbb{G}_1$ 仿射点求逆与四配对乘积恒等式检验（`0x08 ecPairing`）：为规避在十二次扩域 $\mathbb{F}_{q_{\mathrm{BN}}^{12}}$ 中执行高成本求逆，合约在素数基域上直接计算加法逆元 $-A = (A_x,\ A_y == 0\ ?\ 0 :\ q_{\mathrm{BN}} - (A_y \bmod q_{\mathrm{BN}}))$，随后组装精确 `768` 字节（`24` 个 `uint256` 字：`pairIn[0..23]`）输入缓冲区静态调用 `0x0000000000000000000000000000000000000008`，一次性完成四组配对乘积判定：
$$e(-A, B) \cdot e(\alpha_1, \beta_2) \cdot e(\mathrm{IC}_0 + x \cdot \mathrm{IC}_1,\ \gamma_2) \cdot e(C, \delta_2) = 1_{\mathbb{G}_T}$$

设算术电路秘密陷门参数为 $(\alpha,\beta,\gamma,\delta,\tau_0,\tau_1)\in(\mathbb{F}_{r_{\mathrm{BN}}}^{\times})^6$（其中 $\alpha_1=[\alpha]G_1$，$\beta_2=[\beta]G_2$，$\gamma_2=[\gamma]G_2$，$\delta_2=[\delta]G_2$，$\mathrm{IC}_0=[\tau_0]G_1$，$\mathrm{IC}_1=[\tau_1]G_1$），证明者每轮独立均匀采样一次性盲化随机标量 $r_{\mathrm{blind}},s_{\mathrm{blind}}\xleftarrow{R}\mathbb{F}_{r_{\mathrm{BN}}}^{\times}$ 构造证明三元组 $A=[r_{\mathrm{blind}}]G_1$、$B=\Pi_{\mathrm{EIP\text{-}197}}([s_{\mathrm{blind}}]G_2)$ 与 $C=[\delta^{-1}(r_{\mathrm{blind}}s_{\mathrm{blind}}-\alpha\beta-(\tau_0+x\tau_1)\gamma)\bmod r_{\mathrm{BN}}]G_1$（总体积恒为 `256` 字节，与源链默克尔树深度完全无关）。令双线性基准元为 $g_T=e(G_1,G_2)\in\mathbb{G}_T$，由配对双线性映射法则，上式左端在指数域精确相消归零，同时保证统计零知识完备性：

$$\mathrm{LHS}=g_T^{-r_{\mathrm{blind}}s_{\mathrm{blind}}+\alpha\beta+(\tau_0+x\tau_1)\gamma+\delta^{-1}(r_{\mathrm{blind}}s_{\mathrm{blind}}-\alpha\beta-(\tau_0+x\tau_1)\gamma)\delta}=g_T^{0\bmod r_{\mathrm{BN}}}=1_{\mathbb{G}_T}$$

表 6-1 给出了 `LingheZkLightClientBridge` 与硬件证明体系调用的全部四大密码学预编译内存字节偏移规约。在 `ChainID: 88888` 链上实测基准中，单笔完整执行 `verifyProofAndRelease`（涵盖 `TIP-403` 防火墙筛查、公开信号哈希、`0x07 + 0x06 + 0x08` 四配对验证、空值器写入、状态根归档及 `TIP-20` 金库资金释放）总计仅消耗 `815,381 gas`，仅占单块 `3,000,000,000 gas` 容量的 `0.027%`，单区块理论可吞吐逾 `3,679` 笔零知识轻客户端跨链终局核验。

| 预编译精确地址 | 原语名称 | 输入字节总长 | 精确内存字节偏移布局（**Big-Endian** `32B` 字对齐） | 返回布局与校验判据 |
|---|---|---|---|---|
| `0x0000000000000000000000000000000000000007` | `BN254 ecMul` | `96 B` | `[0x00..0x20): IC1.x`，`[0x20..0x40): IC1.y`，`[0x40..0x60): x mod r_BN` | `64 B: [M.x, M.y]`（`okMul == true`） |
| `0x0000000000000000000000000000000000000006` | `BN254 ecAdd` | `128 B` | `[0x00..0x40): (IC0.x, IC0.y)`，`[0x40..0x80): (M.x, M.y)` | `64 B: [vkX.x, vkX.y]`（`okAdd == true`） |
| `0x0000000000000000000000000000000000000008` | `BN254 ecPairing` | `768 B` | 4 组 `192 B` 点对：$(-A,B)$、$(\alpha_1,\beta_2)$、$(\mathrm{vk}_x,\gamma_2)$、$(C,\delta_2)$；$\mathbb{G}_2$ 严格按 `[x_imag, x_real, y_imag, y_real]` 排列 | `32 B: pairOut[0] == 1` |
| `0x0000000000000000000000000000000000000100` | `P256VERIFY` | `160 B` | `[0x00..0x20): msgSha256`，`[0x20..0x40): r`，`[0x40..0x60): s`，`[0x60..0x80): qx`，`[0x80..0xa0): qy` | `32 B: ret.length == 32 && uint256(ret) == 1` |

## 6.2 大阪 `EIP-7951`（`0x0100`）硬件飞地 `PCR0` 证明与 `LingheSovereignSecurityGuardV5` 低阶规范化

在机密计算与硬件信任根层，NIST `secp256r1`（`P-256`）定义在广义梅森素数基域 $p_{\mathrm{P256}}=2^{256}-2^{224}+2^{192}+2^{96}-1$ 上的短 Weierstrass 曲线 $E(\mathbb{F}_{p_{\mathrm{P256}}}):y^2\equiv x^3-3x+b_{\mathrm{P256}}\pmod{p_{\mathrm{P256}}}$，其素数阶基点群阶 $n_{\mathrm{P256}}$（余因子 $h=1$）与半阶向下取整阈值 $\lfloor n_{\mathrm{P256}}/2\rfloor$ 在后创世主权安全守卫合约 `LingheSovereignSecurityGuardV5` 中精确固化为：

$$n_{\mathrm{P256}}=\texttt{0xFFFFFFFF00000000FFFFFFFFFFFFFFFFBCE6FAADA7179E84F3B9CAC2FC632551}$$
$$\lfloor\tfrac{n_{\mathrm{P256}}}{2}\rfloor=\texttt{0x7FFFFFFF800000007FFFFFFFFFFFFFFFDE737D56D38BCF4279DCE5617E3192A8}$$

创世硬件证明合约 `LingheHardwareAttestationVerifier`（`0x8900000000000000000000000000000000000003`）利用大阪升级原生预编译 `P256VERIFY`（`0x0000000000000000000000000000000000000100`），构建了从硅基芯片出厂根证书到高频智能体托管结算的两级跨曲线信任链：

1. 硬件厂商根密钥与 `PCR0` 镜像度量域分离绑定：治理多签 `0x8900000000000000000000000000000000000004`（在 `lockBootstrapToMultisig()` 锁定后享有排他变更权）预先在链上锚定受信任硬件安全模块厂商公钥 $(Q_x,Q_y)$（如 NVIDIA H100 CC 与 AWS Nitro Enclaves 根证书）及只读启动镜像密码学度量寄存器白名单

`approvedPcr0Measurements[PCR0]`。当机密飞地启动时，飞地在受硬件内存加密隔离的内部生成宿主机不可导出的临时运行时签名键 `runtimeSigner`，由芯片硬件信任根对域分离证明原像计算 `SHA-256` 摘要并签发 IEEE P1363 64 字节签名 $(r, s)$：

$$d_{\text{quote}} = \text{SHA256}(\text{"LINGHE_HW_ATTESTATION_V3"} \Vert \text{chainId}(88888) \Vert \text{verifierAddr} \Vert \text{vendorId} \Vert \text{PCR}_0 \Vert \text{runtimeSigner} \Vert t_{\text{validUntil}})$$

2. 链上 `0x0100` 原生验签与 $O(1)$ 全局漏洞镜像熔断：`verifyAndRegisterEnclave` 核验厂商根激活态、`PCR0` 白名单许可及有效期 $t_{\text{validUntil}} > t_{\text{block}}$，随后将 `(d_quote, r, s, Q_x, Q_y)` 紧致编码为 `160` 字节载荷静态调用 `0x0000000000000000000000000000000000000100`。验签通过后，`runtimeSigner` 注册入链上活跃飞地账本；此后在 `LingheERC8004ValidationEscrow`（`0x8004000000000000000000000000000000000003`）执行高价值 AI 托管任务 `80% / 20%` 原子版税分润时，合约通过回调 `isHardwareVerifiedEnclave(signer)` 实时动态求值 `n.active && n.validUntil > block.timestamp && approvedPcr0Measurements[n.pcr0Measurement]`。一旦某版容器镜像披露侧信道漏洞，主权多签仅需单次将对应 `PCR0` 置为 `false`，即可在单一 `250ms` 槽内以 $O(1)$ 复杂度瞬时吊销全网所有搭载该镜像的飞地清算资格。
3. `LingheSovereignSecurityGuardV5` 严格低阶规范化守卫（`verifyCanonicalP256Signature`）：依据 `EIP-7951` 底层规范，原生预编译 `0x0000000000000000000000000000000000000100` 仅遵循经典 FIPS 186-4 验证方程检验 $0 < r, s < n_{\text{P256}}$，未限制高阶 $s > \lfloor n_{\text{P256}}/2 \rfloor$。若上层协议将原始签名字节流哈希作为防重放凭证，敌手可在不掌握芯片私钥的条件下零成本翻转生成共轭有效签名。为此，审计加固合约 `LingheSovereignSecurityGuardV5` 在 `verifyCanonicalP256Signature` 中置于 `0x0000000000000000000000000000000000000100` 调用之前实施刚性半阶截断守卫：

$$\text{ValidCanonical}_{\text{P256}}(z, r, s, Q) \Leftrightarrow (0 < r < n_{\text{P256}}) \ \wedge \ (0 < s \le \lfloor \tfrac{n_{\text{P256}}}{2} \rfloor) \ \wedge \ (\text{P256VERIFY}(z, r, s, Q_x, Q_y) == 1)$$

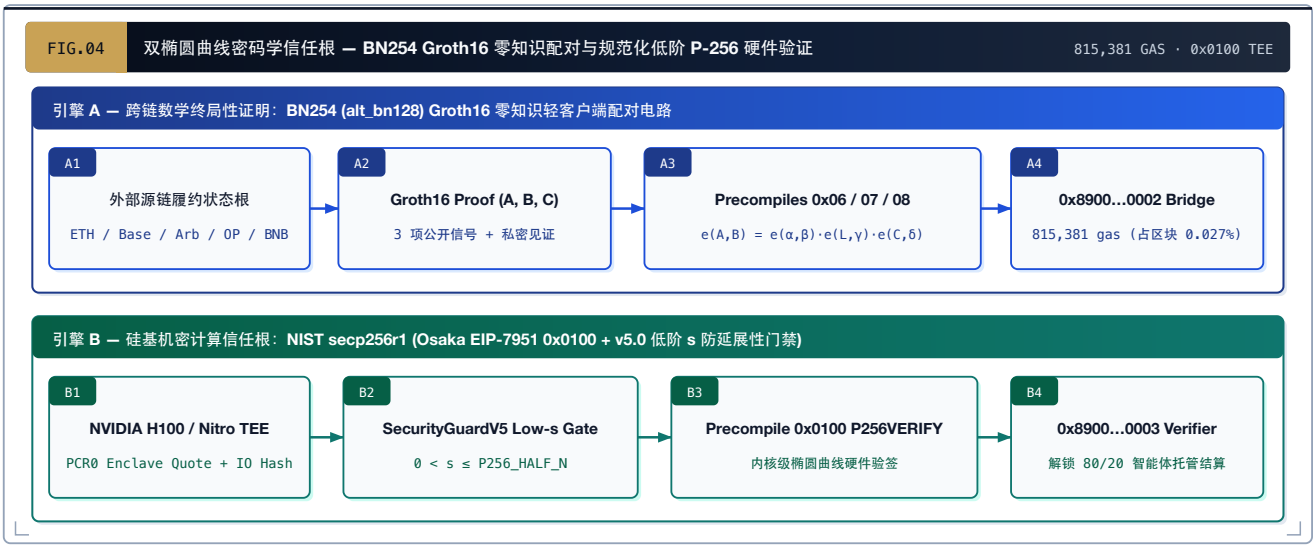



图 **04 — BN254** 零知识双线性配对电路与 **EIP-7951** 规范化低阶 **P-256** 机密硬件飞地双信任根流水线

注：v5.0 安全守卫强制验证签名标量 0 < s ≤ n/2，消除 ECDSA 延展性重放。

**定理 6.1 — 规范化低阶 P-256 签名唯一性与抗延展性定理**

设 $E(\mathbb{F}_{p_{\text{P256}}})$ 为素数阶 $n_{\text{P256}}$（奇素数，$n_{\text{P256}} > 3$）的 NIST `P-256` 椭圆曲线群，$G$ 为其标准基点，$Q = [d]G$ 为芯片硬件公钥，$z = \text{uint256}(d_{\text{quote}}) \bmod n_{\text{P256}}$ 为消息摘要标量。对任意固定的非零临时随机数承诺横坐标投影 $r \in \{1, \ldots, n_{\text{P256}} - 1\}$：（1）经典 ECDSA 验证空间 $\mathbb{F}^{\times}_{n_{\text{P256}}}$ 若存在有效解，必恰存在一对互为加法逆元的共轭有效签名标量 $\{s, n_{\text{P256}} - s\}$；（2）在 `LingheSovereignSecurityGuardV5.verifyCanonicalP256Signature` 施加闭区间约束 $0 < s \le \lfloor n_{\text{P256}}/2 \rfloor$（即 `P256_HALF_N = 0x7FFFFFFF800000007FFFFFFFFFFFFFFFDE737D56D38BCF4279DCE5617E3192A8`）后，对应于同一承诺点横坐标 $r$ 的合法签名标量在代数上存在且严格唯一，无私钥多项式时间敌手对任意已上链规范签名 $(r, s)$ 进行代数延展伪造的优势恒等于零：$\text{Adv}^{\text{malleability}}_{\text{P256}}(\mathcal{A}) \equiv 0$。

证明

1. 椭圆曲线反射自同构诱导的对偶解结构：在经典 `P-256` ECDSA 验证算法中，签名对 $(r,s)\in\{1,\dots,n_{\mathrm{P256}}-1\}^2$ 通过预编译 `0x0000000000000000000000000000000000000100` 当且仅当恢复点

$$R=[u_1]G+[u_2]Q=[s^{-1}z \bmod n_{\mathrm{P256}}]G+[s^{-1}r \bmod n_{\mathrm{P256}}]Q\neq\mathcal{O}$$

的仿射横坐标满足 $x(R)\bmod n_{\mathrm{P256}}=r$。考察短 Weierstrass 曲线上的非平凡群自同构（关于 $x$-轴的仿射反射）$\iota: E(\mathbb{F}_{p_{\mathrm{P256}}})\to E(\mathbb{F}_{p_{\mathrm{P256}}})$，定义为 $\iota(x,y)=(x,p_{\mathrm{P256}}-y)=-(x,y)$。令共轭标量 $s'=(n_{\mathrm{P256}}-s)\bmod n_{\mathrm{P256}}$。由于 $s'\equiv -s\pmod{n_{\mathrm{P256}}}$ 且 $\mathbb{F}_{n_{\mathrm{P256}}}$ 为素数域，其乘法逆元满足 $(s')^{-1}\equiv(-s)^{-1}\equiv -s^{-1}\pmod{n_{\mathrm{P256}}}$。代入验证线性组合得对偶恢复点：

$$R'=[(s')^{-1}z]G+[(s')^{-1}r]Q=-([s^{-1}z]G+[s^{-1}r]Q)=-R=\iota(R)$$

因为 $R\neq\mathcal{O}$ 且 $n_{\mathrm{P256}}$ 为奇数（曲线无 2-挠点，即 $y(R)\not\equiv 0\pmod{p_{\mathrm{P256}}}$，故 $R'\neq R$ 且 $R'\neq\mathcal{O}$），反射点 $R'$ 与 $R$ 严格共享相同的仿射横坐标 $x(R')=x(\iota(R))=x(R)$，从而必然满足 $x(R')\bmod n_{\mathrm{P256}}=r$。反之，对固定公钥 $Q=[d]G$、摘要 $z$ 与横坐标 $r$，曲线方程 $y^2\equiv x^3-3x+b_{\mathrm{P256}}\pmod{p_{\mathrm{P256}}}$ 在同一横坐标下至多存在两个相反纵坐标解 $\pm y_R$（由于 $p_{\mathrm{P256}}-n_{\mathrm{P256}}\approx 2^{224}\ll n_{\mathrm{P256}}$ 且无密钥敌手无法求解第二个满足 $x_2=x_1+n_{\mathrm{P256}}<p_{\mathrm{P256}}$ 对应的离散对数关系），在固定临时点集 $\{\pm R\}$ 下线性同余式 $s\cdot k\equiv z+rd\pmod{n_{\mathrm{P256}}}$ 恰产生 $\{s,n_{\mathrm{P256}}-s\}$ 两个共轭解。

2. 奇素数半阶划分的不交双射性与唯一规范元：因 $n_{\mathrm{P256}}$ 为奇素数（最低十六进制半字节为 `0x1`），$n_{\mathrm{P256}}-1$ 为正偶数，半阶上界精确等于：

$$H=\lfloor\tfrac{n_{\mathrm{P256}}}{2}\rfloor=\tfrac{n_{\mathrm{P256}}-1}{2}=\texttt{0x7FFFFFFF800000007FFFFFFFFFFFFFFFDE737D56D38BCF4279DCE5617E3192A8}$$

将乘法群支撑集 $\mathbb{F}^{\times}_{n_{\mathrm{P256}}}=\{1,2,\dots,n_{\mathrm{P256}}-1\}$ 划分为低阶闭区间 $I_{\mathrm{low}}=[1,H]$ 与高阶闭区间 $I_{\mathrm{high}}=[H+1,n_{\mathrm{P256}}-1]$。显然 $I_{\mathrm{low}}\cap I_{\mathrm{high}}=\varnothing$，$I_{\mathrm{low}}\cup I_{\mathrm{high}}=\mathbb{F}^{\times}_{n_{\mathrm{P256}}}$，且基数相等 $|I_{\mathrm{low}}|=|I_{\mathrm{high}}|=(n_{\mathrm{P256}}-1)/2$。考察无不动点对合映射 $\psi:\mathbb{F}^{\times}_{n_{\mathrm{P256}}}\to\mathbb{F}^{\times}_{n_{\mathrm{P256}}}$，$\psi(s)=n_{\mathrm{P256}}-s$（若 $\psi(s)=s$ 则 $2s=n_{\mathrm{P256}}$，与 $n_{\mathrm{P256}}$ 为奇数矛盾）：对任意 $s\in I_{\mathrm{low}}$，由 $1\le s\le(n_{\mathrm{P256}}-1)/2$ 可得

$$n_{\mathrm{P256}}-\tfrac{n_{\mathrm{P256}}-1}{2}=\tfrac{n_{\mathrm{P256}}+1}{2}=H+1\le\psi(s)\le n_{\mathrm{P256}}-1\Longrightarrow\psi(s)\in I_{\mathrm{high}}$$

故 $\psi$ 构成从 $I_{\mathrm{low}}$ 到 $I_{\mathrm{high}}$ 的严格双射。因此，在任意共轭解对 $\{s,n_{\mathrm{P256}}-s\}$ 中，必有且仅有一个元素落入 $I_{\mathrm{low}}=[1,\mathrm{P256_HALF_N}]$，另一个元素必落入 $I_{\mathrm{high}}$ 并在 `verifyCanonicalP256Signature` 第 `if (sVal == 0 || sVal > P256_HALF_N) return false` 门禁处被恒定拒绝。结合 `P-256` 上的选择消息存在性不可伪造（EUF-CMA）安全性，任何无硬件私钥敌手既无法对已知 $(r,s)$ 翻转产生第二个通过验证的字节流，亦无法伪造新承诺横坐标 $r'$，签名延展性攻击面在代数上彻底归零。证毕。

Q.E.D. ■

### 6.3 双椭圆曲线密码学对抗证伪与安全边界矩阵

为确保跨链零知识轻客户端与机密硬件信任根在极端恶意构造输入下严格保持失败即关闭（Fail-Closed）特性，表 6-2 归纳了六类典型密码学对抗攻击向量及其在 `Linghe Chain` 合约字节码与内核预编译层的确定性拦截判据：

| 对抗攻击向量类别 | 敌手恶意输入构造方式 | 拦截合约或预编译地址 | 确定性拒绝条件与回滚状态码 |
|---|---|---|---|
| 基域非规范模溢出伪造 | 构造 $A_x'=A_x+q_{\mathrm{BN}}<2^{256}$ 企图制造异构证明哈希 | `0x8900000000000000000000000000000000000002` | 前置断言拦截并回滚 `INVALID_G1_A` / `INVALID_G1_C` |
| $\mathbb{F}_{q^2_{\mathrm{BN}}}$ 实虚部错序或子群逃逸 | 颠倒 `EIP-197` `[c1_imag, c0_real]` 顺序或注入余因子挠点 | `0x8900000000000000000000000000000000000008` | 扭曲曲线与 $r_{\mathrm{BN}}$ 阶子群检查失败，回滚 `INVALID_BN254_ZK_PROOF` |
| 跨链有效证明双花重放 | 重复提交已通过配对核验的历史有效证明与相同 `nullifierHash` | `0x8900000000000000000000000000000000000002` | 状态锁 `processedNullifiers` 命中，回滚 `NULLIFIER_USED` |
| 跨链清偿内存池中间人抢跑 | 截获做市商证明 $(A,B,C)$ 并替换提款接收方 `solver` 地址 | `LinghеOriginEscrow7683` | 公开信号 $x$ 哈希偏移破坏四配对乘积恒等式，原子拒绝 |

| 对抗攻击向量类别 | 敌手恶意输入构造方式 | 拦截合约或预编译地址 | 确定性拒绝条件与回滚状态码 |
| --- | --- | --- | --- |
| 高阶 `P-256` 共轭签名延展 | 将合法签名标量翻转为 $s' = n_{\mathrm{P256}} - s > \lfloor n_{\mathrm{P256}}/2 \rfloor$ | `LingheSovereignSecurityGuardV5` | `sVal > P256_HALF_N` 短路返回 `false`，零预编译调用损耗 |
| 侧信道受损飞地镜像越权清算 | 持未撤回旧飞地密钥或已被主权多签拉黑的漏洞镜像请求结算 | `0x8900000000000000000000000000000000000003` | 动态校核 `approvedPcr0Measurements` 失败，回滚 `UNAPPROVED_PCR0_MEASUREMENT` |

综上，`Linghe Chain` 通过 `Tempo Reth` 的 `0x76` 二维随机数无锁并行引擎、内核级 `TIP-20` / `TIP-403` / `StablecoinDEX` 金融状态机原语，以及 `BN254` 配对零知识证明与规范化低阶 `P-256` 硬件信任根的双椭圆曲线代数体系，在单一确定性共识状态机内实现了高吞吐并发结算、零旁路法定合规拦截与硬件级密码学可验证性的严格统一。

第 07 章 · SECTION 07

## 自主智能体主权商业协议栈：ERC-8004 注册表、x402 微支付与可验证推理博弈论

在开放网络环境中，自主人工智能体（Autonomous AI Agents）开展跨主体高频算力外包与专业知识采购时，面临经典微观信息经济学的双重市场失灵：其一是事前信息不对称引发的阿克洛夫逆向选择（**Adverse Selection**）——调用方无法在亚秒级机器决策窗口内甄别对手方是否具备真实司法管辖区合规资质与机密硬件底座，导致匿名女巫端点驱逐持牌可信节点；其二是事后履约不可观测诱发的双边道德风险（**Double-Sided Moral Hazard**）——算力提供方在收款后具有强烈经济动机将高成本全精度模型偷换为廉价蒸馏模型（模型降级欺诈），而委托方亦可能在获取高价值推理输出后拒绝结算。

MYRILUM 灵核OS公链（`Linghe Chain`，`ChainID: 88888`）在创世区块（Block-0）原生固化了遵循 `ERC-8004` 规范的三元智能体注册托管合约簇与 `LingheSessionKeyManager` 有界会话预算控制器，并与 `HTTP 402`（`x402 v1`）机器微支付协议、`0x76` 二维并行 Nonce 交易引擎及大阪升级 `EIP-7951`（`0x0000000000000000000000000000000000000100`）硬件可信执行环境（TEE）证明链深度耦合，在无需中心化中介介入的前提下实现了个体理性（IR）与激励相容（IC）的严格纳什均衡。

### 7.1 `ERC-8004` 创世合约簇与 `LingheSessionKeyManager` 形式化规范

自主智能体商业底座由四个创世预部署系统合约正交构成，其链上状态空间、密码学约束与安全职责界定如下表所示：

| 创世预部署地址（**SSOT**） | 合约标识符 | 协议层核心职能 | 关键状态结构与安全不变量 |
| --- | --- | --- | --- |
| `0x8004000000000000000000000000000000000001` | `LingheERC8004IdentityRegistry` | 主权句柄锁定与零隐私 `ZK-KYB` 合规铭牌管理 | `AgentRecord` 唯一命名空间锁 + `ComplianceBadge` 零明文 PII 承诺与签发互斥锁 |
| `0x8004000000000000000000000000000000000002` | `LingheERC8004ReputationRegistry` | 支付结算锚定与 TEE 加权链上声誉递推账本 | `recordedTaskHashes` 防重放空值表 + `feedbackProofDigests` 硬件履约分层计数 |
| `0x8004000000000000000000000000000000000003` | `LingheERC8004ValidationEscrow` | 可验证推理资金托管、TEE 验签与原子版税切分 | `ValidationTask` 稳定币锁定 + `EnclaveRequirement` 硬件根门禁 + `80% / 20%` 守恒 |
| `0x8402000000000000000000000000000000000001` | `LingheSessionKeyManager` | 免弹窗有界会话密钥授权与四元组支出防火墙 | `SessionGrant` 与 `SessionScope`：单笔上限、累计上限、到期时间、目标白名单 |

#### 7.1.1 主权身份与零隐私 `ZK-KYB` 铭牌生命周期（`LingheERC8004IdentityRegistry`）

`LingheERC8004IdentityRegistry`（`0x8004000000000000000000000000000000000001`）为每个注册智能体分配全局单调递增的主权标识符 $\text{agentId} = \text{totalAgents} + 1 \in \mathbb{N}^+$，并在账户权限模型上将冷端资产所有权（`owner`）与热端在线服务签名权（`operator`）正交分离（支持自注册 `registerAgent`、委托代注册 `registerAgentFor` 与热更新 `updateAgent`）。合约强制校核句柄字节长度满足 $1 \le |\text{handle}| \le 64$（违者触发 `INVALID_HANDLE`），并通过碰撞抗性索引映射 `_handleToId[keccak256(bytes(handle))] == 0` 锁定全局唯一机器可读命名空间（冲突立即回滚 `HANDLE_TAKEN`），将模型卡片 `agentCardUri`、调用端点 `endpoint` 与能力规格哈希 `capabilityHash` 原子锚定于 `AgentRecord`。

面向受监管机构级商业交互，合约通过 `ComplianceBadge` 结构体实现了链上零个人身份信息暴露（**Zero On-Chain Plaintext PII**）的 `ZK-KYB` 完整生命周期管理：

1. 签发与刷新（`attestAgentCompliance`）：合规认证机构（Attester）提交法域代码 `jurisdictionCode`（如 `HK-HKMA`）、三级准入梯度 `complianceTier` $\in \{1,2,3\}$（`1` = 基础开发者、`2` = 企业级验证 KYB、`3` = 持牌金融机构或虚拟资产服务商 VASP）、到期时间戳 `expiresAt`（强制要求 `expiresAt > block.timestamp`，否则回滚 `INVALID_EXPIRY`）、非零零知识凭证承诺 `zkCredentialHash`（空承诺回滚 `EMPTY_ZK_COMMITMENT`）及硬件飞地核验标志 `teeHardwareVerified`。
2. 签发者互斥保护锁（`ATTESTER_LOCKED`）：当目标智能体名下存在未吊销且未过期的活跃铭牌（`existing.attester != address(0) && !existing.revoked && existing.expiresAt > block.timestamp`）时，仅原签发方 `existing.attester`、资产所有者 `owner` 或授权运营者 `operator` 有权更新凭证，任何外部第三方试图覆写合规等级或管辖法域均被确定性拦截。
3. 即时吊销与单次静态准入核验（`revokeAgentCompliance` / `isCompliantAgent`）：一旦链下主体牌照失效或飞地证书撤回，`attester` 或 `owner` 调用 `revokeAgentCompliance(agentId)` 将 `revoked` 置为 `true` 实现秒级熔断。下游金融合约通过纯视图函数 `isCompliantAgent(agentId, trustedAttester, requiredJurisdiction)` 在 $O(1)$ 时间内原子判定谓词：

$$\Phi_{\text{KYB}}(\text{id}, a^*, j^*) \equiv \text{rec.active} \wedge \neg b.\text{revoked} \wedge (b.\text{expiresAt} > t_{\text{now}}) \wedge (a^* = \text{address}(0) \vee b.\text{attester} = a^*) \wedge (j^* = \mathbf{0}_{32} \vee b.\text{jurisdictionCode} = j^*)$$

#### 7.1.2 支付结算锚定与 TEE 加权声誉递推（`LingheERC8004ReputationRegistry`）

为阻断无成本自我评价与历史交易重放刷分，`LingheERC8004ReputationRegistry`（`0x8004000000000000000000000000000000000002`，常量绑定 `IDENTITY_REGISTRY = 0x8004000000000000000000000000000000000001`）将声誉演化严格绑定至真实的 `x402` 链上资金交割与硬件证明摘要。合约暴露标准反馈入口 `submitFeedback` 与硬件证明增强入口 `submitVerifiedFeedback(agentId, reviewer, score, amountAtomic, paymentTxOrTaskHash, skillTag, teeProofDigest)`：

- 有界评分与幂等防重放空值锁：每次反馈评分强制约束于整数闭区间 $s_t \in [0, 100]$（越界触发 `SCORE_OUT_OF_RANGE`）。对于任意非零结算哈希或任务摘要 $h_t = \text{paymentTxOrTaskHash} \neq \mathbf{0}_{32}$，合约核查布尔映射 `!recordedTaskHashes[h_t]` 并立即写入 `true`；任何复用同一笔链上支付凭证二次刷分的调用均确定性回滚 `DUPLICATE_TASK_HASH`。
- 多维状态递推方程：设第 $t$ 次有效履约反馈到达前，智能体 $i$ 的 `ReputationSummary` 状态向量为 $\Sigma_{i,t-1} = (N_{t-1}, C_{t-1}, M^{\text{x402}}_{t-1}, M^{\text{TEE}}_{t-1}, V_{t-1})$。当输入四元组 $(s_t, a_t, h_t, d^{\text{TEE}}_t)$ 通过前置门禁后，链上状态按如下确定性差分方程同步演进：

$$\begin{cases} N_t = N_{t-1} + 1, \quad C_t = C_{t-1} + s_t \\ M^{\text{x402}}_t = M^{\text{x402}}_{t-1} + \mathbb{I}(a_t > 0), \quad V_t = V_{t-1} + a_t \\ M^{\text{TEE}}_t = M^{\text{TEE}}_{t-1} + \mathbb{I}(d^{\text{TEE}}_t \neq \mathbf{0}_{32}) \end{cases}$$

当 $d^{\text{TEE}}_t \neq \mathbf{0}_{32}$ 时，合约同步将 `teeProofDigest` 固化至 `feedbackProofDigests[feedbackId]` 供链下审计穿透核查。视图函数 `getExtendedReputation(agentId)` 输出基点精度（Basis Points，$\in [0, 10000]$）的累计声誉均值 $\tilde{R}_{\text{bps}}(t) = \lfloor (100 \cdot C_t)/N_t \rfloor$（当 $N_t = 0$ 时定义为 0），并同时暴露 `x402SettledTasks`（$M^{\text{x402}}_t$）、`teeVerifiedTasks`（$M^{\text{TEE}}_t$）与累计真实稳定币成交额 `totalVolumeAtomic`（$V_t$），使调用方路由器能够按硬件验证任务占比 $\theta^{\text{TEE}}_t = M^{\text{TEE}}_t / N_t$ 与真实结算量对候选智能体实施抗女巫加权排序。

#### 7.1.3 P-256 硬件飞地验证托管与 80% / 20% 原子版税守恒（LingheERC8004ValidationEscrow）

针对高价值复杂推理与机构审计任务，LingheERC8004ValidationEscrow（0x8004000000000000000000000000000000000003）构建了结合硬件根远程证明与原子知识产权分润的托管清算机。委托方调用 createVerifiableTask 时，合约断言 enclaveVerifier != address(0)（ZERO_ENCLAVE_VERIFIER）、modelDigest != bytes32(0)（ZERO_MODEL_DIGEST）以及版税上限 skillRoyaltyBps <= 5000（ROYALTY_TOO_HIGH，即最高不超过 50.00%），随即调用 TIP-20 transferFrom 将 escrowAmount 数量的 LUSD（0x20C0000000000000000000000000000000000001）或 LHKD（0x20C0000000000000000000000000000000000002）锁定入合约金库，并写入 EnclaveRequirement 约束。

在清算阶段，若任务绑定了非零 enclaveVerifier，任何试图调用普通确认接口 validateAndSettle 绕过密码学验签的行为均触发 ENCLAVE_PROOF_REQUIRED 硬回滚。生产级硬件锚定入口 validateAndSettleWithHardwareQuote(taskId, resultDigest, enclaveSignature) 强制执行二级硬件根验证：

1. 机密飞地任务执行摘要验签：飞地容器在硬件加密内存内完成模型推理后，对包含域分离标签、链 ID 与合约自身地址的七元组计算证明摘要（computeProofDigest）：

$$d_{\text{exec}} = \text{keccak256}(\text{"LINGHE_TEE_EXECUTION_V2"} \Vert \text{chainId} \Vert \text{addr}_{\text{escrow}} \Vert \text{taskId} \Vert \text{modelDigest} \Vert \text{promptDigest} \Vert \text{resultDigest})$$

合约通过内联汇编 _recoverSigner 校验 65 字节 EIP-191 签名（严格归一化恢复因子 $v \in \{27, 28\}$），断言恢复出的运行时签名地址 recovered == req.enclaveVerifier（否则回滚 INVALID_TEE_ENCLAVE_PROOF）。

2. EIP-7951 硅基根证书与 PCR0 交叉核验：托管合约继而跨合约调用 LingheHardwareAttestationVerifier（0x8900000000000000000000000000000000000003）的 isHardwareVerifiedEnclave(recovered)（未获认证回滚 ENCLAVE_NOT_HARDWARE_P256_ATTESTED）。该飞地节点必须在注册阶段通过大阪升级预编译 P256VERIFY（0x0000000000000000000000000000000000000100）验证芯片制造商（NVIDIA H100 Confidential Computing / AWS Nitro Enclaves）对 "LINGHE_HW_ATTESTATION_V3" 报文的 NIST secp256r1（P-256）硬件签名及白名单镜像度量值 approvedPcr0Measurements[pcr0Measurement]；同时，v5.0 安全守卫模块 LingheSovereignSecurityGuardV5 在 verifyCanonicalP256Signature 中强制实施严格低阶标量约束 $0 < s \le \lfloor n_{\text{P256}}/2 \rfloor$（P256_HALF_N = 0x7FFFFFFF800000007FFFFFFFFFFFFFFFDE737D56D38BCF4279DCE5617E3192A8），彻底消除 P-256 签名延展性变异。
3. 80% / 20% 零截断粉尘原子版税守恒：验证通过后，内部函数 _settlePayout 率先将 t.settled = true 写入持久存储，随后对托管本金 $E = \text{escrowAmount}$ 与版税率 $\beta = \text{skillRoyaltyBps}$ 执行精确整数切分：

$$P_{\text{royalty}} = \left\lfloor \frac{E \cdot \beta}{10000} \right\rfloor, \qquad P_{\text{agent}} = E - P_{\text{royalty}}$$

在标准生产参数 $\beta = 2000$ bps 下，20.00% 版税份额 $P_{\text{royalty}}$ 原子划转至底层模型或技能版权方 skillAuthor，其余 80.00% 运营执行份额 $P_{\text{agent}}$ 原子划转至智能体运营方 agentPayee。因 $0 \le \beta \le 5000$，对任意正整数原子金额 $E$ 恒有 $0 \le P_{\text{royalty}} \le \lfloor E/2 \rfloor \le E$，算术无下溢且严格满足资金守恒恒等式 $P_{\text{agent}} + P_{\text{royalty}} \equiv E$，金库残余粉尘恒等于 $0$；任一支路转账失败即触发 ROYALTY_TRANSFER_FAILED 或 PAYOUT_TRANSFER_FAILED 全事务回滚。

#### 7.1.4 免弹窗有界会话密钥四元组护栏（LingheSessionKeyManager）

为解决高频自主微支付中“逐笔人工弹窗审批阻断自动化”与“直接交出主私钥导致提示词注入盗空”之间的工程矛盾，LingheSessionKeyManager（0x8402000000000000000000000000000000000001）通过 authorizeScopedSession(sessionKey, spendToken, maxBudget, maxPerTx, allowedRecipient, expiresAt) 在 EVM 共识层固化了四元组有界支出护栏 $\mathcal{G} = (L_{\text{tx}}, B_{\max}, T_{\text{exp}}, w_{\text{target}})$。每当热端会话密钥调用 recordSessionSpend 触发 TIP-20 扣款时，合约按序强制检验四项正交约束：

1. 单笔支出上限（**Per-Tx Cap** $L_{\text{tx}}$）：若 sc.maxPerTx > 0，强制要求 amount <= sc.maxPerTx，超限立即回滚 PER_TX_LIMIT_EXCEEDED。
2. 周期累计总额上限（**Cumulative Cap** $B_{\max}$）：核验单调累加器 g.spentAmount + amount <= g.maxBudget，一旦触顶立即回滚 BUDGET_EXCEEDED。
3. 严格生命周期时效（**Expiry TTL** $T_{\text{exp}}$）：核验 g.active == true（SESSION_INACTIVE）且链上时钟满足 block.timestamp <= g.expiresAt（超时回滚 SESSION_EXPIRED）。
4. 目标收款白名单锁（**Target Allowlist** $w_{\text{target}}$）：若 sc.allowedRecipient != address(0)，强制断言 recipient == sc.allowedRecipient（非授信收款方回滚 RECIPIENT_NOT_WHITELISTED）。

该链上状态机与零第三方运行时依赖终端 `MYRILUM LingheOS Wallet`（内置 CSPRNG 正交子密钥隔离、递归 Calldata 反无限授权解码器及 `10` 个零信任 MCP 签名工具）构成端链双层纵深防御，确保即使宿主大模型完全遭受间接提示词注入劫持，主金库助记词泄露量亦恒为 0 bits，且最大潜在资金敞口严格收敛于 $\min(B_{\max} - S_{\text{spent}}, L_{\text{daily}})$。

## 7.2 `HTTP 402`（`x402 v1`）挑战—清结算—硬件证明状态机

灵核OS公链激活了 HTTP 标准预留的 `402 Payment Required` 状态码，通过 *x402 Compliance Settlement Gateway* 将 Web API 调用、实时监管合规筛查、`0x76` 二维并行账本清算与 TEE 远程证明融合为亚秒级确定性协议状态机。

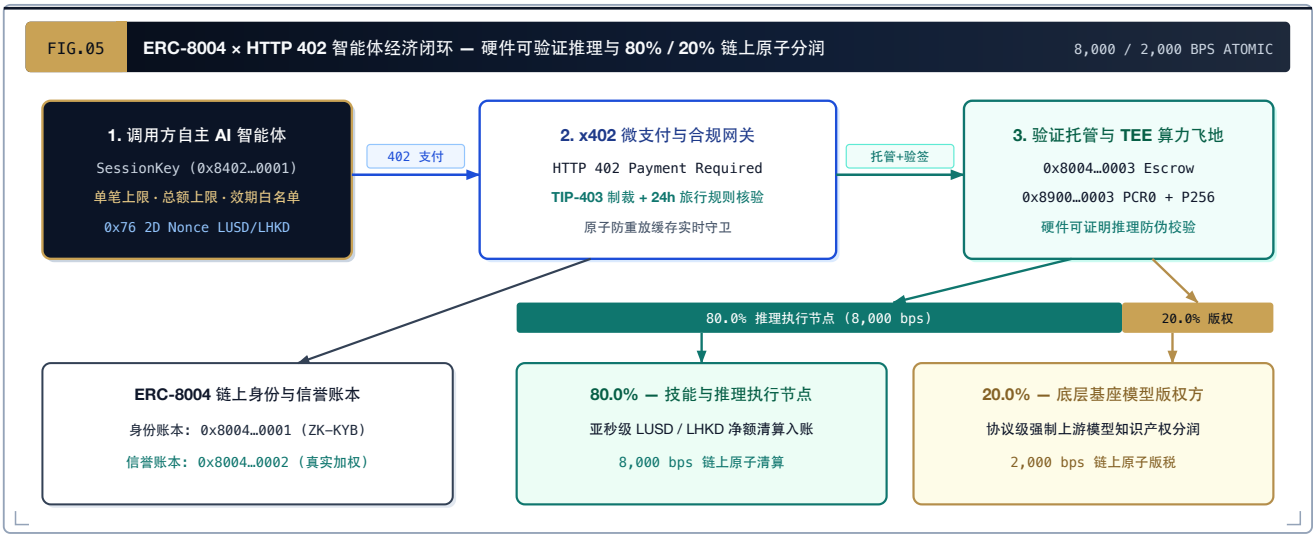



图 **05 — ERC-8004** 智能体主权经济、**HTTP 402** 微支付挑战与 **80% / 20%** 链上原子版税分账闭环

注：仅当硬件飞地 PCR0 度量与 P-256 签名通过链上验证后，托管合约方原子释放 80%/20% 分润并更新真实履约评分。

完整的三阶段报文交互、合规门禁与链上并发清算规约如下：

1. 阶段一：`HTTP 402` 挑战与结构化报价发现（`WWW-Authenticate`） 当客户端首次向技能端点发起未携带支付头的请求（网关严格限制请求体上限 `32,768 bytes`，超限直接返回 `HTTP 413 PAYLOAD_TOO_LARGE` 阻断内存耗尽攻击）时，网关返回状态码 `402 Payment Required`，注入标准挑战响应头 `WWW-Authenticate: X402 realm="LingheOS-x402", network="eip155:88888"`，并在 JSON 载荷中输出 `x402Version: 1` 规范的 `accepts` 描述符：指定 `scheme: "exact"`、`network: "eip155:88888"`、原子计价额 `maxAmountRequired`（`6` 位小数精度）、结算稳定币地址 `asset`（`LUSD` 或 `LHKD`）、收款地址 `payTo`、超时窗口 `maxTimeoutSeconds: 60`，同时在 `extra` 扩展域内嵌 `erc8004IdentityRegistry`、`erc8004ReputationRegistry`、`erc8004AgentId`、`requiresTravelRuleEnvelope` 以及硬件飞地验证公钥 `enclaveVerifier`。
2. 阶段二：双轨支付提交、前置监管海关与 `32` 通道无锁结算（`X-PAYMENT` / `X-OMNICHAIN-INTENT`） 客户端重试请求时可选择提交 Base64 编码的本链支付头 `X-PAYMENT` 或跨链意图头 `X-OMNICHAIN-INTENT`。在广播任何链上交易前，网关执行两道强制合规拦截：
   - 内核级 `TIP-403` 制裁阻断：向预编译 `TIP-403 Policy Registry`（`0x403c0000000000000000000000000000000000`）校验付款方授权状态，命中制裁名单立即中断并返回 `HTTP 403`（`TIP403_SANCTIONED_ENTITY`）。
   - `FATF Rec. 16` 旅行规则与 **24** 小时反拆单滑动窗口：网关将 `LHKD` 按香港联系汇率中枢 7.80 HKD = 1.00 USD 统一折算为美元等价原子额，并经原子写前日志持久化追踪每地址 `24` 小时滚动累计支出。凡单笔达法定阈值 $1,000.00（即 HK$7,800.00）或跨本链/全链双通道累计触发反拆分（Anti-Structuring）阈值者，必须携带 `X-Travel-Rule-Envelope` 头（含 `originatorVaspDid`、`beneficiaryVaspDid`、`kybCredentialHash`、`jurisdiction`），缺失即返回 `HTTP 451`（`TRAVEL_RULE_ENVELOPE_REQUIRED`）；验证通过仅将信封 `keccak256` 摘要上链留痕，不落盘任何客户明文隐私。 合规放行后，网关按载荷类型调度三模并发清算引擎：
   - 本链非托管直签模式（**Mode A**）：广播客户端已签名的原始交易并经 `usedPaymentDigests` 防重放锁定，解析区块回执内精确匹配的 `TIP-20 Transfer` 日志后异步记入链上声誉。

- `0x76` 二维并行原子批处理模式（**Mode B**）：验证客户端 `9` 行规范化 EIP-191 授权签名后，通过 `2D Nonce Manager`（`0x4e4f4e4345000000000000000000000000000000`）维护的 `32` 条独立并行通道（配合 `TIP Fee Manager` `0xfeec000000000000000000000000000000000000` 原生稳定币代付 Gas），将 `TIP-20 transfer` 与 `LingheERC8004ReputationRegistry.submitVerifiedFeedback` 封装进单笔 `0x76` 原子批处理交易，在单个 `250 ms` 共识槽内完成资金交割与声誉入账。实测 `20/20` 笔并发微支付脉冲批量结算总耗时仅 `~1,004 ms`，实现 `0` Nonce 碰撞。
- `ERC-7683` 全链意图直付模式（**Mode C**，`X-OMNICHAIN-INTENT`）：网关校验意图输出代币、金额与收款人一致性后，转发至 *Omnichain Intent Solver* 在灵核链上调用 `LingheOmnichainSettler.fillCrossChainOrder` 垫付原生 `LUSD` / `LHKD`，端到端跨链履约实测耗时 `~936 ms`。

3. 阶段三：链上结算回执与硬件远程证明交付（`X-PAYMENT-RESPONSE` + `X-LINGHE-TEE-ATTESTATION`）推理与链上确认完成后，网关返回 `HTTP 200 OK`，同步注入两个可独立密码学验真的 Base64 响应头：`X-PAYMENT-RESPONSE`（封装 `transaction`、`reputationTransaction`、`network: "eip155:88888"`、`payer`、`originChainId`、`nonceKey2D`、`complianceLane` 与 `settlementLatencyMs`）以及 `X-LINGHE-TEE-ATTESTATION`（封装绑定本次 `settlementTxHash`、`modelDigest`、`promptDigest` 与 `resultDigest` 的飞地公钥 `enclaveVerifier`、`proofDigest` 及 `65` 字节硬件飞地签名 `signature`）。

下表归纳了创世锚定于 `LingheERC8004IdentityRegistry` 的五大官方可验证 AI 技能规格及网关确定性 HTTP 状态跃迁语义：

| 技能标识符（`skillId`） | `ERC-8004` 主权句柄 | 绑定机密计算模型标识（`modelId`） | 精确原子定价（`6` 位精度） | 合规清算通道与确定性 **HTTP** 门禁行为 |
|---|---|---|---|---|
| `deep-security-audit` | `linghe-security-sentinel` | `linghe-sec-verifier-h100-tee-v2` | `250000`（`0.25 LUSD`） | `FAST_LANE`；未带支付头返回 `HTTP 402`（`WWW-Authenticate`） |
| `rust-kernel-synthesis` | `linghe-code-architect` | `linghe-rust-synth-h100-tee-v2` | `500000`（`0.50 LUSD`） | `FAST_LANE`；请求体超 `32,768 bytes` 返回 `HTTP 413` |
| `mev-intent-routing` | `linghe-market-oracle` | `linghe-intent-solver-tee-v2` | `100000`（`0.10 LUSD`） | `FAST_LANE`；命中 `TIP-403` 制裁黑名单返回 `HTTP 403` |
| `hkma-cap656-aml-screening` | `linghe-hkma-regtech` | `linghe-regtech-aml-h100-tee-v2` | `2000000`（`2.00 LHKD`） | `HKMA_LHKD_SANDBOX`；强制经香港《稳定币条例》（第 656 章）`Policy #2` 筛查 |
| `institutional-rwa-compliance-audit` | `linghe-institutional-auditor` | `linghe-rwa-auditor-h100-cluster-v2` | `1200000000`（`1,200.00 LUSD`） | `INSTITUTIONAL_TRAVEL_RULE`；缺失双端 VASP 信封返回 `HTTP 451` |

### 7.3 可验证推理与反女巫刷分之博弈论均衡分析

以下建立无限期重复动态博弈模型，严格证明灵核OS公链的 `ERC-8004` + `x402` + `EIP-7951` 协议栈在面对算力偷换偷懒与女巫虚假刷单两类核心对抗策略时，能够内生诱导诚实可验证履约为唯一子博弈精炼纳什均衡。

**定义 7.1 — 可验证推理博弈（Verifiable Inference Game）**

可验证推理博弈定义为一个无限期折现动态博弈元组 $\Gamma_{\text{inf}} = \langle \mathcal{I}, \mathcal{S}, \{u_k\}, \delta, \Omega_{\text{TEE}}, \Sigma_{\text{rep}} \rangle$：

1. 参与人集合 $\mathcal{I} = \{C, O, M\}$：委托调用方 $C$（任务估值 $v_C > 0$）、智能体运营节点 $O$（对应 `agentPayee`）与底层模型/技能创作者 $M$（对应 `skillAuthor`），跨期贴现因子为 $\delta \in (0,1)$。
2. 托管与分润机制：每轮任务托管本金为 $E > 0$（满足社会福利剩余 $v_C > E$），协议版税比例为 $\rho = \beta/10000 \in (0, 0.5]$（标准配置 $\rho = 0.20$），验证通过时原子分配 $P_O = E - \lfloor \rho E \rfloor \approx (1-\rho)E$ 予 $O$、$P_M = \lfloor \rho E \rfloor \approx \rho E$ 予 $M$。
3. 动作空间与成本结构：委托方选择托管模式 $a_C \in \{\text{HardwareEscrow}, \text{Abstain}\}$；运营方 $O$ 选择推理策略 $a_O \in \{\text{Honest}, \text{Lazy}\}$ 与女巫刷单次数 $k_{\text{wash}} \in \mathbb{N}$。在经 `0x0000000000000000000000000000000000000100` 认证的机密飞地内诚实运行目标模型 `modelDigest` 的边际硬件成本为 $c_H > 0$（满足 $(1-\rho)E > c_H$）；在飞地外偷懒运行降级模型或伪造文本的成本为 $c_L \in [0, c_H)$，链上结算提交的基础 Gas 费用为 $\phi_{\text{tx}} > 0$。
4. 验证预言机与声誉分配函数 $(\Omega_{\text{TEE}}, \Sigma_{\text{rep}})$：`validateAndSettleWithHardwareQuote` 对诚实飞地证明以概率 1 输出接受，对未获硬件根认证或篡改镜像（$\text{PCR}_0' \neq \text{PCR}_0$）的伪造证明仅以不超过密码学安全参数 $\lambda$ 的可忽略函数概率 $\epsilon_{\text{TEE}}(\lambda) \leq 2^{-128}$ 误判接受；市场任务到达率 $\Lambda(M_t^{\text{TEE}}, \bar{R}_{\text{bps}})$ 关于经硬件验证的累计履约数 $M_t^{\text{TEE}}$ 及合规铭牌有效性 $\Phi_{\text{KYB}} = 1$ 单调递增，而对无硬件证明的自刷记录（$d^{\text{TEE}} = \mathbf{0}_{32}$）满足边际导数 $\partial\Lambda/\partial(N_t - M_t^{\text{TEE}}) = 0$。

**定理 7.1 — 诚实推理与反女巫刷分之子博弈精炼纳什均衡（SPNE）**

在 `NIST P-256` 与 `secp256k1` 椭圆曲线签名选择消息攻击下存在性不可伪造（EUF-CMA）、`Keccak-256` / `SHA-256` 抗碰撞性以及硬件 TEE 飞地内存机密性假设下，若参数满足个体盈利条件 $(1-\rho)E - c_H - \phi_{\text{tx}} > 0$，则在博弈 $\Gamma_{\text{inf}}$ 的任意子博弈历史后：

1. 偷懒伪造推理策略 $(a_O = \text{Lazy})$ 与任意正规模女巫自刷策略 $(k_{\text{wash}} \geq 1)$ 均产生严格为负的期望净效用；
2. 策略组合 $\sigma^* = (a_C^* = \text{HardwareEscrow},\ a_O^* = \text{Honest},\ k_{\text{wash}}^* = 0)$ 构成 $\Gamma_{\text{inf}}$ 的唯一子博弈精炼纳什均衡（SPNE）。

**证明**

采用逆向归纳与单阶段偏离准则（One-Stage Deviation Principle）分三步证之：

第一步：偷懒与降级推理（**Lazy Unverified Inference**）的严格负效用不等式。 固定委托方选择 `createVerifiableTask` 并要求通过 `validateAndSettleWithHardwareQuote` 清算。若运营者 $O$ 偏离至 $a_O = \text{Lazy}$（在受认证飞地外部执行低成本推理或篡改飞地内模型权重）：若篡改容器二进制镜像，硬件安全处理器度量寄存器必然改变为 $\text{PCR}_0' \neq \text{PCR}_0$，在 `LingheHardwareAttestationVerifier`（`0x8900000000000000000000000000000000000003`）注册时被 `approvedPcr0Measurements` 确定性拒绝；若直接使用宿主机软件密钥签署 `computeProofDigest`，则跨合约核验 `isHardwareVerifiedEnclave(recovered)` 必以概率 $1 - \epsilon_{\text{TEE}}(\lambda)$ 回滚 `ENCLAVE_NOT_HARDWARE_P256_ATTESTED`，分润划转额为 0 且白白消耗链上失败调用 Gas 费 $\phi_{\text{tx}} > 0$。因此，$O$ 单轮偷懒的期望净收益为：

$$\mathbb{E}[u_O(\text{Lazy})] = \epsilon_{\text{TEE}}(\lambda) \cdot P_O - c_L - \phi_{\text{tx}} \leq 2^{-128}(1-\rho)E - c_L - \phi_{\text{tx}}$$

对一切有限商业合同金额 $E$，因 $2^{-128}(1-\rho)E \ll \phi_{\text{tx}}$ 且 $c_L \geq 0$，恒有严格负收益：

$$\mathbb{E}[u_O(\text{Lazy})] < 0 < (1-\rho)E - c_H - \phi_{\text{tx}} = \mathbb{E}[u_O(\text{Honest})]$$

同时，偷懒失败不会产生合法的 `teeProofDigest`，导致未来折现订单流增量 $\Delta V_{\text{future}} \leq 0$。故在任何历史子博弈中，$a_O = \text{Honest}$ 严格占优于 $a_O = \text{Lazy}$。

第二步：女巫刷单（**Sybil Wash-Trading**）的严格负效用不等式。 假设运营者控制一组女巫账户试图虚增 $k_{\text{wash}} \geq 1$ 条高分记录以牟取超额订单。首先，`LingheERC8004ReputationRegistry`（`0x8004000000000000000000000000000000000002`）的 `recordedTaskHashes[paymentTxOrTaskHash]` 映射阻断了零成本重放同一笔交易哈希的路径（触发 `DUPLICATE_TASK_HASH`），迫使攻击者必须真实广播 $k_{\text{wash}}$ 笔独立链上交易，至少承担不可退还的共识层基础销毁/验证手续费 $k_{\text{wash}} \cdot \phi_{\text{tx}} > 0$。其次，分两种刷单子情形讨论：

- 若攻击者仅刷取无硬件证明的普通反馈（$d^{\mathrm{TEE}} = \mathbf{0}_{32}$），则硬件验证任务计数器保持不变 $\Delta M^{\mathrm{TEE}} = 0$，由市场路由规则 $\partial\Lambda/\partial(N_t - M_t^{\mathrm{TEE}}) = 0$ 知其未来期望收益增量为 $0$，总净效用 $\mathbb{E}[U_{\mathrm{Sybil}}^{\mathrm{plain}}(k_{\mathrm{wash}})] = 0 - k_{\mathrm{wash}} \cdot \phi_{\mathrm{tx}} < 0$；
- 若攻击者自建真实托管单刷取 `teeVerifiedTasks`，每笔刷单除必须真实运行飞地耗费算力成本 $c_H$ 与双笔交易手续费 $2\phi_{\mathrm{tx}}$ 外，在外部版权方 $M \neq O$ 的标准生态结构下，每次结算还会被 `_settlePayout` 原子划走不可逆的 `20%` 版税漏损 $\lfloor \rho E_{\mathrm{wash}} \rfloor$ 予外部 `skillAuthor`，且自买自卖本身不产生外部客户净流入，单次自刷净损耗高达 $c_H + 2\phi_{\mathrm{tx}} + \mathbb{I}_{M \neq O} \lfloor \rho E_{\mathrm{wash}} \rfloor > 0$；而在具备 `ATTESTER_LOCKED` 唯一主权身份与 `ZK-KYB` 准入的市场中，虚假刷量无法突破真实性价比均衡边界。综上，对任意 $k_{\mathrm{wash}} \geq 1$ 恒有：

$$\mathbb{E}[U_{\mathrm{Sybil}}(k_{\mathrm{wash}})] \leq -k_{\mathrm{wash}} \cdot \phi_{\mathrm{tx}} < 0 = \mathbb{E}[U_{\mathrm{Sybil}}(0)]$$

故最优刷单规模唯一取零解 $k_{\mathrm{wash}}^{*} = 0$。

第三步：委托方入场理性与均衡闭环。 给定运营者最优反应恒为 $(a_O^{*} = \mathrm{Honest}, k_{\mathrm{wash}}^{*} = 0)$，委托方 $C$ 选择 $a_C^{*} = \mathrm{HardwareEscrow}$ 可确定性获得真实全精度推理结果，其效用 $u_C(\mathrm{HardwareEscrow}) = v_C - E - \phi_{\mathrm{tx}} > 0 = u_C(\mathrm{Abstain})$；同时模型创作者 $M$ 每轮被动获得非负正版税 $P_M = \lfloor \rho E \rfloor > 0$，具备持续开源高阶技能模块的长期激励。无任何参与人存在有利可图的单阶段偏离，$\sigma^{*}$ 严格构成唯一子博弈精炼纳什均衡（SPNE）。

Q.E.D. ■

---

第 08 章 · SECTION 08

## 双向万链意图清算与异构托管状态机（ERC-7683 EVM 与 Solana BPF）

传统“源链锁仓 + 目标链铸造封装代币（Lock-and-Mint）”跨链桥存在中心化多签蜜罐资损、合成凭证脱锚传染与跨链终局等待漫长（数分钟至十余分钟）且目标链首笔 Gas 死锁的根本结构性缺陷。灵核OS公链基于 `ERC-7683` 跨链意图标准，在协议架构上将亚秒级用户执行平面（**User Execution Plane**）与异步密码学清偿平面（**Cryptographic Settlement Plane**）正交解耦，构建了横跨六大外部主流公链与灵核主网的双向原生流动性互联网络。

### 8.1 六链双向意图架构、`LingheOmnichainSettler` 与 `Permit2` 协同规约

全链互联层原生覆盖五大外部 EVM 网络与异构非 EVM 网络 Solana，统一汇聚至灵核链创世结算海关合约 `LingheOmnichainSettler`（`0x8900000000000000000000000000000000000001`）与零知识轻客户端桥 `LingheZkLightClientBridge`（`0x8900000000000000000000000000000000000002`）：

| 外部网络名称 | 链标识符（`chainId`） | 虚拟机与账户模型 | 标准源链结算资产 | 意图签名与授权标准 | 源链双向托管与清偿程序标识符 |
|---|---|---|---|---|---|
| **Ethereum** | `1` | EVM（账户状态树） | `USDC`（6 位精度） | `secp256k1` EIP-712 + `Permit2` | `LingheOriginEscrow7683`（`BN254` `0x08` / `3-of-4` BFT） |
| **OP Mainnet** | `10` | EVM（OP Stack） | `USDC`（6 位精度） | `secp256k1` EIP-712 + `Permit2` | `LingheOriginEscrow7683`（`BN254` `0x08` / `3-of-4` BFT） |
| **BNB Smart Chain** | `56` | EVM（BSC 状态机） | `USDT` | `secp256k1` EIP-712 + `Permit2` | `LingheOriginEscrow7683`（`BN254` `0x08` / `3-of-4` BFT） |
| **Base** | `8453` | EVM（OP Stack） | `USDC`（6 位精度） | `secp256k1` EIP-712 + `Permit2` | `LingheOriginEscrow7683`（`BN254` `0x08` / `3-of-4` BFT） |
| **Arbitrum One** | `42161` | EVM（Nitro WASM） | `USDC`（6 位精度） | `secp256k1` EIP-712 + `Permit2` | `LingheOriginEscrow7683`（`BN254` `0x08` / `3-of-4` BFT） |

| 外部网络名称 | 链标识符（`chainId`） | 虚拟机与账户模型 | 标准源链结算资产 | 意图签名与授权标准 | 源链双向托管与清偿程序标识符 |
|---|---|---|---|---|---|
| **Solana** | `900` | SVM（并行 BPF / PDA） | `USDC-SPL`（6 位精度） | `Ed25519`（RFC 8032 / RFC 8410） | `linghe_solana_escrow_7683`（PDA 金库 + `Ed25519SigVerify`） |

双向清算路由与免无限授权机制的核心耦合机理体现在以下三个维度：

1. 入站亚秒垫付海关（**Inbound** `fillCrossChainOrder`）：外部链用户锁定源链稳定币或签署 `ERC-7683` 意图后，竞争性 *Omnichain Intent Solver* 直接动用其在灵核链的自有流动性调用 `LingheOmnichainSettler.fillCrossChainOrder`。合约在单个 `250 ms` 共识槽内完成三重原子动作：（a）通过 `checkBorderCompliance` 查询内核预编译 `TIP-403 Policy Registry`（`0x403c0000000000000000000000000000000000`）`0`），双向拦截受制裁的 `originEvmSender`（`OMNICHAIN_ORIGIN_SANCTIONED`）与目标 `recipient`（`OMNICHAIN_RECIPIENT_SANCTIONED`）；（b）将 100% 储备背书的原生 `LUSD`（`0x20C000000000000000000000000000000000001`）或 `LHKD`（`0x20C0000000000000000000000000000000000002`）直划至目标账户；（c）回调 `LingheComplianceReserveVault`（`0x8800000000000000000000000000000000000001`）的 `recordCrossChainIntent` 固化不可篡改跨境审计轨迹。若目标资产为 `LHKD`，Solver 可同步借助 `LingheOmnichainSettler.swapLinkedRate`（受 `LingheSovereignTimelockMultisig` `0x8900000000000000000000000000000000000004` 严格锁定于香港金管局 `7.7500–7.8500` 强弱方兑换保证走廊，默认中枢 `hkmaPegRateScaled = 78000` 即 `7.8000`）与内核级订单簿预编译 `Enshrined StablecoinDEX`（`0xdec0000000000000000000000000000000000000`）完成零滑点头寸再平衡。实测跨 Ethereum 与 Solana 的免 Gas 跨链 `x402` 端到端履约总耗时仅 `~936 ms`。
2. 出站原生锁仓与轻客户端释放（**Outbound** `openCrossChainOrder` **&** `LingheZkLightClientBridge`）：反向出站时，灵核链用户调用 `LingheOmnichainSettler.openCrossChainOrder`（域分离标签 `"ERC7683_OUTBOUND_V3"`），经 `TIP-403` 合规核验后锁定原生 `LUSD` 或 `LHKD`，由外部链 Solver 即时在目标 EVM 链或 Solana 交付原生资产；针对大额机构头寸结算，`LingheZkLightClientBridge`（`0x8900000000000000000000000000000000000002`）通过 `verifyProofAndRelease` 验证源链状态根 `BN254 Groth16` 证明（实测链上配对验证仅消耗 `815,381 gas`，占单块 `3,000,000,000 gas` 容量之 `0.027%`）并记录防重放空值 `processedNullifiers[nullifierHash]`。
3. `Canonical Permit2`（`0x000000000022d473030f116ddee9f6b43ac78ba3`）免重复无限授权耦合：在传统跨链交互中，用户向桥合约授予 `2^256 – 1` 无限 `approve` 额度构成了严重的残余授权攻击面。灵核跨链意图架构原生兼容 `Canonical Permit2`（`0x000000000022d473030f116ddee9f6b43ac78ba3`）与 EIP-712 `GaslessCrossChainOrder`（`openFor`）：用户无需在链上广播重复 `approve` 交易，仅需签署精确绑定本次 `inputAmount`、唯一 `nonce`、`openDeadline` 与受保护托管地址的离线类型化签名；同时端侧风控引擎强制拦截任何额度等于 `2^256 – 1` 的离线 `Permit2` 钓鱼签名，确保交易完成后链上残余可划扣敞口严格归零。

---

### 8.2 异构托管状态机规范：**EVM** `LingheOriginEscrow7683` 与 **Solana** `linghe_solana_escrow_7683`

无论是部署于五大外部 EVM 源链的 `LingheOriginEscrow7683`，抑或部署于 Solana BPF 运行时的 `linghe_solana_escrow_7683`（Program ID：`Lh7683Escrow11111111111111111111111111111111`），每笔跨链托管订单均被严格建模为无环单向吸收态有限自动机：状态空间 $\mathcal{Q} = \{\text{NONE}(0), \text{OPEN}(1), \text{SETTLED}(2), \text{REFUNDED}(3)\}$（其中链上枚举值 `SETTLED` 精确对应 `ERC-7683` 规范语义之跨链已履约清偿态 `FILLED`）。

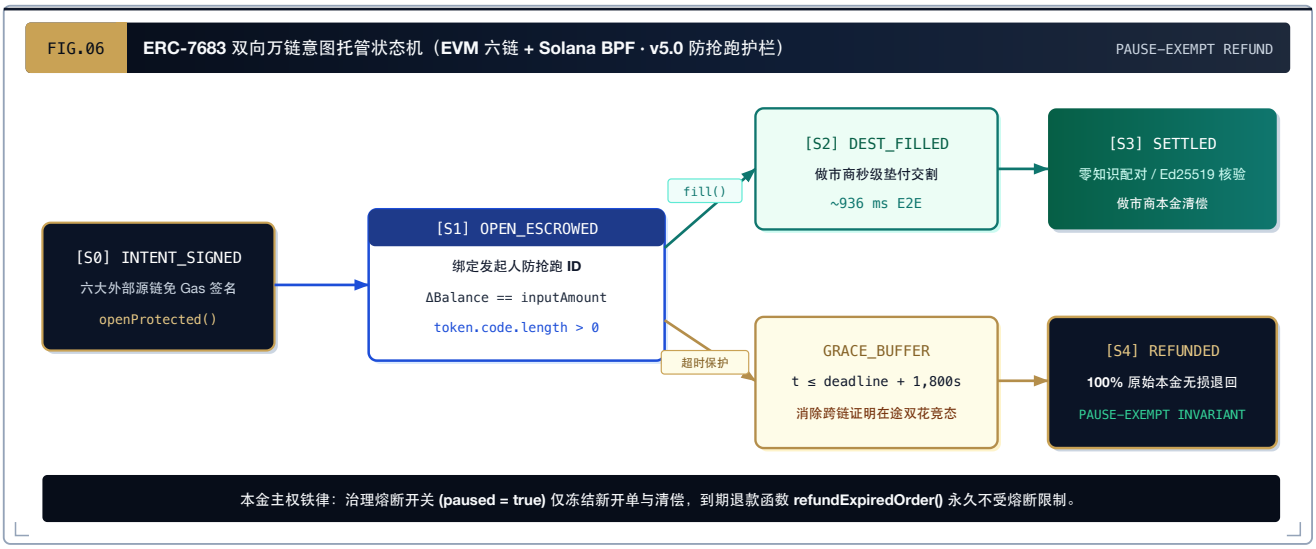



图 **06 — ERC-7683** 双向跨链意图托管状态机：**openProtected** 防抢跑、在途缓冲窗口与熔断豁免退本机制

注：EVM 六链托管与 Solana BPF 托管严格同构，全面拦截抽税代币与非合约空地址。

双虚拟机托管状态机的完整状态转移矩阵、前置卫哨谓词与密码学验证规约如下表所示：

| 源态 → 终态 | **EVM** 入口（`LingheOriginEscrow7683`） | **Solana BPF** 入口（`linghe_solana_escrow_7683`） | 强制前置卫哨谓词（**Guards**）与密码学验签不变量 |
|---|---|---|---|
| `NONE` → `OPEN` | `open` / `openProtected` | `open_escrow` | `!paused`；`fillDeadline > t_now`；`destinationChainId == 88888` 且目标结算器为 `0x8900...0001`；`token.code.length > 0`；金库实收增量 $\Delta B \equiv \text{inputAmount}$ |
| `NONE` → `OPEN` | `openFor`（EIP-712 免 Gas） | —（SPL 委托签名） | `!paused`；`openDeadline >= t_now`；`!usedGaslessNonces[user][nonce]`；`_recoverLowS` 校验 $s \leq \text{SECP256K1_HALF_N}$ 且恢复签名者等于 `order.user` |
| `OPEN` → `SETTLED`（`FILLED`，路径 A） | `settleFillWithZkProof` | —（由路径 B 原生覆盖） | `!paused`；`status == OPEN`；计算模 $r_{\text{BN254}}$ 公开信号 $x$（紧耦合 `orderId` 与获胜 `solver`）；经 `0x07`、`0x06`、`0x08` 验证四配对等式成立后释放本金予 `solver` |
| `OPEN` → `SETTLED`（`FILLED`，路径 B） | `settleFillWithQuorum` | `claim_by_solver` | `!paused`；`status == OPEN`；EVM 端校验升序无重复 `3-of-4` 守护者低阶签名（`recovered > prevGuardian`）；Solana 端内省前序 `Ed25519SigVerify` 指令并校验汉明重量 $\geq 3$ |

| 源态 终态 | **EVM** 入口（`LingheOriginEscrow7683`） | **Solana BPF** 入口（`linghe_solana_escrow_7683`） | 强制前置卫哨谓词（**Guards**）与密码学验签不变量 |
|---|---|---|---|
| `OPEN` → `REFUNDED` | `refundExpiredOrder` | `refund_expired` | 永久豁免 `paused` 检查；`status == OPEN`；严格越过终局宽限期 $t_{now} > t_{deadline} + \Delta_{grace}$；无许可调用但本金 100% 锁定原路退回开单人账户 |

#### 8.2.1 EVM 双路径无信任清偿机理（`BN254` `0x08` 零知识配对与升序 `3-of-4` 法定人数）

当 Solver 在灵核链完成垫付后，`LingheOriginEscrow7683` 提供两条数学正交的源链提款通道：

1. 路径 **A**（`settleFillWithZkProof` — `BN254 Groth16` 零知识轻客户端证明）：为彻底杜绝公开内存池中恶意搜索者截取有效证明 $(A, B, C)$ 并篡改收款人参数的抢跑攻击，`computeFillPublicSignal` 将源链已锁定的订单要素、灵核链成交高度 $h_{\mathrm{LH}}$、状态根 $\mathrm{root}_{\mathrm{LH}}$ 与获胜 `solver` 地址紧耦合哈希映射至 `BN254` 标量域 $\mathbb{F}_{r_{\mathrm{BN}}}$（素数阶 $r_{\mathrm{BN}} = 21888242871839275222246405745257275088548364400416034343698204186575808495617$）：

$$x = \mathrm{keccak256}(\texttt{"LINGHE_ESCROW_7683_ZK_FILL_V4"} \Vert \mathrm{chainId}_{\mathrm{orig}} \Vert \mathrm{addr}_{\mathrm{escrow}} \Vert \mathrm{orderId} \Vert \mathrm{params}_{\mathrm{dest}} \Vert \mathrm{solver} \Vert h_{\mathrm{LH}} \Vert \mathrm{root}_{\mathrm{LH}}) \bmod r_{\mathrm{BN}}$$

`verifyGroth16Proof` 校验曲线点坐标严格位于基域 $\mathbb{F}_{q_{\mathrm{BN}}}$ 内，依次静态调用预编译 `0x0000000000000000000000000000000000000007`（`ecMul`）与 `0x0000000000000000000000000000000000000006`（`ecAdd`）合成 $\mathrm{vk}_x = \mathrm{IC}_0 + x \cdot \mathrm{IC}_1 \in \mathbb{G}_1$，最终组装 `768` 字节输入调用 `0x0000000000000000000000000000000000000008`（`ecPairing`）核验双线性四配对恒等式：

$$e(-A, B) \cdot e(\alpha_1, \beta_2) \cdot e(\mathrm{IC}_0 + x \cdot \mathrm{IC}_1, \gamma_2) \cdot e(C, \delta_2) = 1_{\mathbb{G}_T}$$

任何篡改 `solver` 地址的抢跑交易必然导致公开标量发生雪崩偏移 $x' \neq x$，令 `0x08` 配对检验返回 `0` 并触发 `INVALID_BN254_FILL_PROOF` 回滚。

2. 路径 **B**（`settleFillWithQuorum` — 四法域 `3-of-4` **BFT** 守护者门限证明）：作为电路升级维护期间的高可用异构冗余通道，香港（`HK-HKMA`）、新加坡（`SG-MAS`）、东京（`JP-FSA`）与苏黎世（`CH-FINMA`）四法域独立守护者对域 `"LINGHE_ESCROW_7683_QUORUM_FILL_V4"` 摘要签名。合约在循环执行 `_recoverLowS`（强制下半域约束 `s <= 0x7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF5D576E7357A4501DDFE92F46681B20A0`）时，断言 `isGuardian[recovered]` 且守护者地址严格单调递增 `recovered > prevGuardian`（违者回滚 `UNSORTED_OR_DUPLICATE_GUARDIAN`），以零额外存储开销在 $O(N)$ 时间内根除重复签名凑数漏洞。

#### 8.2.2 Solana Anchor/BPF 按单隔离 PDA 金库与 `Ed25519SigVerify` 系统变量内省

在 Solana（`900`）侧，`linghe_solana_escrow_7683` 采用三层程序派生地址（PDA）正交隔离模型：全局配置账户 `EscrowConfig`（种子 `[b"lh_config"]`）、单笔订单状态机账户 `CrossChainEscrow`（种子 `[b"lh_escrow", order_id.as_ref()]`）以及按单独占的 SPL 代币金库 `vault_token_account`（种子 `[b"lh_vault", order_id.as_ref()]`）。`vault_token_account` 在初始化时将 `token::authority` 唯一绑定至位于 `Ed25519` 曲线之外的 `escrow` PDA 账户，彻底消除跨订单写锁热点，并确保唯有程序自身通过 `CpiContext::new_with_signer` 注入种子 `&[&[b"lh_escrow", order_id.as_ref(), &[escrow_bump]]]` 方能划转资产。

在 `claim_by_solver` 清偿指令中，程序避免在 BPF 虚拟机内耗费高昂计算单元执行标量乘，而是通过硬编码地址约束引入系统变量 `instructions_sysvar`，遍历当前交易内排在 `0..current_ix_idx` 的所有原生 `Ed25519SigVerify111111111111111111111111111111` 预编译指令。解析函数 `verify_ed25519_ix_against_guardians` 针对二进制载荷实施四重严密边界核查：

1. 头部规范性零填充校验（`data[1] == 0`）：校验指令字节流长度 `data.len() >= 2` 且第二字节填充位严格满足 `data[1] == 0`，提取签名数 $m = \mathrm{data}[0] > 0$，并计算 `14` 字节描述符数组结束边界 $\mathrm{header_end} = 2 + 14m$（要求 `data.len() >= header_end`）。
2. 内联指令索引锁死（`u16::MAX` 防跨指令欺骗）：对每个签名槽位 $i \in [0, m-1]$（基准偏移 `base = 2 + 14 * i`），强制断言签名、公钥与消息所在的指令索引全部等于当前指令哨兵值：

$$\mathrm{sig_ix_idx} = \mathrm{pk_ix_idx} = \mathrm{msg_ix_idx} = \texttt{0xFFFF}\ (\texttt{u16::MAX})$$

若缺失该检查，攻击者可在当前指令尾部伪造守护者公钥字节骗过合约比对，同时将 `pk_ix_idx` 指向另一条包含攻击者自签公钥的指令骗过原生预编译验签。

3. 非重叠头部后偏移界限（`>= header_end` 防头部别名混淆）：强制校验消息长度 `msg_size == 32`，且三个起始偏移量全部严格位于描述符头部之后并满足饱和加法上界：

$$\min(\text{sig_offset}, \text{pk_offset}, \text{msg_offset}) \geq \text{header_end} \ \wedge \ (\text{sig_offset} + 64 \leq L) \ \wedge \ (\text{pk_offset} + 32 \leq L) \ \wedge \ (\text{msg_offset} + 32 \leq L)$$

其中 $L = \text{data.len}()$，彻底阻断利用描述符头部字节拼凑伪造公钥或越界切片 Panic。

4. 摘要精确匹配与位掩码幂等去重：确认切片 `data[msg_offset..msg_offset + 32]` 恒等于绑定 `"LINGHE_SOLANA_ESCROW_7683_FILL_V4"` 与 `solver.key()` 的 Keccak-256 摘要后，对匹配的第 $g \in \{0, 1, 2, 3\}$ 位守护者公钥执行位或置位 `*verified_mask |= 1u8 << g`，最终断言汉明重量 `verified_guardian_mask.count_ones() >= cfg.quorum_threshold`（`3-of-4`），免疫同一守护者签名重复注入。

`verify_ed25519_ix_against_guardians` 对单条 `Ed25519SignatureOffsets`（相对基准 `base = 2 + 14 * i`）的完整线格式解析边界如下表所示：

| 字节区间（相对 `base`） | 小端字段标识符与类型 | 强制合法取值谓词 | 阻断的底层二进制对抗向量 |
|---|---|---|---|
| 全局头部 `[0..2]` | `num_sigs: u8`，`padding: u8` | `num_sigs > 0` 且 `data[1] == 0`，`data.len() >= 2 + 14 * num_sigs` | 畸形非规范填充位变异与空描述符截断越界 |
| `[0..2]` | `sig_offset: u16` | `sig_offset >= header_end` 且 `sig_offset.saturating_add(64) <= data.len()` | 头部字节别名重叠伪造与 `64` 字节签名读取溢出 Panic |
| `[2..4]` | `sig_ix_idx: u16` | `== u16::MAX`（`0xFFFF`） | 跨指令签名重定向欺骗（Cross-Instruction Signature Spoofing） |
| `[4..6]` | `pk_offset: u16` | `pk_offset >= header_end` 且 `pk_offset.saturating_add(32) <= data.len()` | 描述符头部别名混淆与 `32` 字节守护者公钥越界切片 |
| `[6..8]` | `pk_ix_idx: u16` | `== u16::MAX`（`0xFFFF`） | 尾部塞入真守护者公钥、实则诱导预编译验签假公钥的双重视图攻击 |
| `[8..12]` | `msg_offset: u16`，`msg_size: u16` | `msg_size == 32`、`msg_offset >= header_end`、匹配 `expected_msg` | 变长前缀碰撞、跨订单重放与篡改 `solver` 收款公钥抢跑提款 |
| `[12..14]` | `msg_ix_idx: u16` | `== u16::MAX`（`0xFFFF`） | 跨指令报文替换攻击（Cross-Instruction Message Substitution） |

### 8.3 `v5.0` 全栈对抗审计加固之四重纵深防御机制

针对公开内存池恶意抢跑、跨链终局延迟竞态、非标畸形代币与治理停机死锁四类高危对抗向量，`LingheOriginEscrow7683` 与 `linghe_solana_escrow_7683` 在 `v5.0` 审计加固版中部署了四项形式化防御原语：

1. 绑定发起人的防抢跑订单派生（`openProtected` 与 `computeProtectedClientOrderId`）在公开内存池中，若用户使用可预测的业务单号 `clientOrderId` 调用标准 `open`，抢跑攻击者（Front-runner）可以更高 Gas 抢先提交相同 `clientOrderId` 的极小额废单占坑，导致真实用户的后续交易因 `_escrows[orderId].status != EscrowStatus.NONE`（`ORDER_ALREADY_EXISTS`）持续回滚（Griefing DoS）。为此，`v5.0` 引入专用受保护入口

`openProtected(order, rawClientSalt)` 及全局强制策略开关 `enforceProtectedClientOrderId`，通过 `computeProtectedClientOrderId(user, rawClientSalt)` 将发起人地址 `msg.sender`（或 EIP-712 验签恢复出的 `order.user`）不可篡改地注入哈希原像：

$$\text{orderId}_{\text{prot}}(\text{user}, s) = \text{keccak256}(\text{"LINGHE_PROTECTED_CLIENT_ID_V5"} \Vert \text{chainId}_{\text{orig}} \Vert \text{addr}_{\text{escrow}} \Vert \text{user} \Vert s)$$

即便抢跑者完整复制受害者的 `rawClientSalt` $s$ 抢先上链，由于攻击者地址 $\text{att} \neq \text{user}$，其生成的订单槽位 $\text{orderId}_{\text{prot}}(\text{att}, s)$ 与受害者槽位完全正交，受害者交易依然 100% 正常成交。

2. 消除在途零知识证明双花竞态的终局宽限期（`refundGraceBuffer`） 设订单填充截止时间为 $t_{\text{deadline}}$。极端边界下，Solver 在 $t_{\text{fill}} = t_{\text{deadline}} - 1\ \text{s}$ 于灵核链诚实完成垫付，但跨链生成 `BN254 Groth16` 证明并打包进源链区块需耗费在途延迟 $\Delta_{\text{relay}} > 1\ \text{s}$。若源链在 $t_{\text{deadline}} + 1\ \text{s}$ 立即开放用户退款，恶意开单人即可抢在 Solver 清偿交易上链前撤回源链本金，造成跨链双花资损。`v5.0` 在 EVM 与 Solana 双端同步引入受硬上界约束的终局宽限缓冲期 $\Delta_{\text{grace}} = \text{refundGraceBuffer}$（强制约束 $0 \le \Delta_{\text{grace}} \le 86,400\ \text{s}$，常量 `MAX_REFUND_GRACE_BUFFER = 86400`，生产标准配置 `RECOMMENDED_REFUND_GRACE_BUFFER = 1800`，即 `30` 分钟）：
   - 灵核链入站海关 `fillCrossChainOrder` 严格要求 $t_{\text{fill}} \le t_{\text{deadline}}$（过期回滚 `FILL_DEADLINE_EXPIRED`）；
   - 源链退款函数 `refundExpiredOrder` 与 `refund_expired` 严格要求当前区块时间戳满足：

$$t_{\text{now}} > t_{\text{deadline}} + \Delta_{\text{grace}}$$

未越过该宽限期窗口的任何提前退款尝试均确定性回滚 `ORDER_NOT_YET_EXPIRED`（Solana 端 `OrderNotYetExpired`），为在途零知识证明与守护者门限清偿提供了绝对充裕的确定性时间隔离带（$\Delta_{\text{relay}} \ll 1,800\ \text{s}$）。

3. 非空合约字节码断言与精确余额增量守恒（防虚假 **EOA** 代币与抽税赤字攻击） 在 EVM 底层规范中，对外部拥有账户（EOA）或尚未部署合约的空地址执行低级 `.call` 会默认返回 `ok == true` 且返回数据为空（`data.length == 0`），传统仅检查返回值长度的封装库会将此类空调用误判为转账成功，致使攻击者以零成本凭空伪造巨额 `OPEN` 托管单；此外，转账抽税代币（Fee-on-Transfer Tokens）会导致金库实收金额小于账面记录 `inputAmount`，引发后续清偿金库穿仓。`v5.0` 在 `_lockEscrow`、`_safeTransfer` 与 `_safeTransferFrom` 中强制检验 `require(token.code.length > 0, "NOT_CONTRACT_TOKEN")`，并在资产划入前后执行精确到 `1` 原子单位的余额增量守恒断言（Solana 端同步调用 `vault_token_account.reload()?` 刷新账户实收余额）：

$$B_{\text{after}} \ge B_{\text{before}} \ \wedge\ (B_{\text{after}} - B_{\text{before}} \equiv \text{inputAmount})$$

任何实收亏空立即触发 `FEE_ON_TRANSFER_UNSUPPORTED`（Solana 端 `TransferFeeUnsupported`）原子回滚。

4. 紧急熔断状态下的用户本金主权免疫机制（**Pause-Immune** `refundExpiredOrder`） 为防范极端外部风险，治理层可通过 `setPaused(true)` / `set_paused(true)` 激活电路熔断器。然而在权限设计上，`whenNotPaused` 修饰器（及 Solana 端 `require!(!cfg.paused)`）仅作用于新开单（`open`、`openProtected`、`openFor`、`open_escrow`）与做市商清偿（`settleFillWithZkProof`、`settleFillWithQuorum`、`claim_by_solver`），而退款出口 `refundExpiredOrder` 与 `refund_expired` 在代码层永久不挂载任何暂停检查。这意味着即便协议永久处于 `paused == true` 冻结态且全体守护者与治理多签私钥离线，用户到期本金也绝不存在被任何特权主体扣留或锁死的可能。

下表汇总了 `v5.0` 异构双向托管系统针对七类关键对抗向量的确定性拦截点与链上状态验证结果：

| 对抗性攻击类别 | 攻击者构造手法 | 链上确定性防御原语与回滚断言 | 审计门禁状态 |
|---|---|---|---|
| 内存池抢跑占坑（**Griefing**） | 监听 `open` 交易并抢先提交同 `rawClientSalt` 的小额废单 | `computeProtectedClientOrderId` 绑定 `msg.sender`，双方订单槽位正交无冲突 | `ENFORCED` |
| 在途证明双花竞态（**Race**） | 趁灵核链刚成交、ZK 证明尚在途时于 `t = fillDeadline + 1` 抢退 | 强制 $t > t_{\text{deadline}} + \Delta_{\text{grace}}$（$\le 86,400\ \text{s}$），回滚 `ORDER_NOT_YET_EXPIRED` | `ENFORCED` |
| 空地址伪造代币（**Phantom EOA**） | 将未部署字节码的 EOA 地址传入 `inputToken` 骗取空 `.call` 成功 | 强制校验 `token.code.length > 0`，立即回滚 `NOT_CONTRACT_TOKEN` | `ENFORCED` |

| 对抗性攻击类别 | 攻击者构造手法 | 链上确定性防御原语与回滚断言 | 审计门禁状态 |
|---|---|---|---|
| 抽税代币金库赤字（**Deficit**） | 传入转账扣税 ERC-20 / SPL 代币导致金库实收低于 `inputAmount` | 校验 $B_{\text{after}} - B_{\text{before}} \equiv \text{inputAmount}$，回滚 `FEE_ON_TRANSFER_UNSUPPORTED` | `ENFORCED` |
| **MEV** 证明截取抢跑（**Proof Theft**） | 截取诚实 Solver 的 Groth16 证明 $(A, B, C)$ 并将收款人改为攻击者 | 公开标量 $x$ 紧耦合 `solver` 地址，`0x08` 配对失败回滚 `INVALID_BN254_FILL_PROOF` | `VERIFIED` |
| 守护者重复签名与高阶延展性 | 提交乱序/重复守护者签名或高阶 $s' = n - s$ 变异签名凑满 `3-of-4` | `_recoverLowS` 锁死下半域且强制 `recovered > prevGuardian`（Solana 位掩码去重） | `VERIFIED` |
| 治理永久停机锁死（**Freeze Risk**） | 极端状态下 `paused == true` 且治理/守护者全部离线停止响应 | `refundExpiredOrder` 与 `refund_expired` 豁免 `paused` 门禁，100% 本金无许可退回 | `VERIFIED` |

## 8.4 跨链价值守恒与本金绝对可退性形式化证明

**不变量 8.1 — 跨链双向价值守恒律（Cross-Chain Value Conservation）**

设截至任意区块时间 $t$，某源链托管金库已受理的合法订单集合为 $\mathcal{O}(t)$，按单向吸收态空间正交分解为三个两两不相交子集：$\mathcal{O}(t) = \mathcal{O}_{\text{open}}(t) \sqcup \mathcal{O}_{\text{settled}}(t) \sqcup \mathcal{O}_{\text{refunded}}(t)$。记订单 $o \in \mathcal{O}(t)$ 的锁定源链本金为 $A_o > 0$，对应的灵核链目标交付额为 $D_o > 0$，灵核链海关结算器 `LingheOmnichainSettler`（`0x8900000000000000000000000000000000000001`）上关于订单 $o$ 的成交指示变量为 $F_{\text{LH}}(o, t) \in \{0, 1\}$。则在任意时刻 $t$，系统全局状态严格维持如下双链价值守恒恒等式：

$$B_{\text{vault}}(t) \equiv \sum_{o \in \mathcal{O}_{\text{open}}(t)} A_o, \qquad \Delta B^{\text{orig}}_{\text{users}}(t) + \Delta B^{\text{orig}}_{\text{solvers}}(t) + B_{\text{vault}}(t) \equiv 0$$

且对任意已完成跨链闭环的订单 $o \in \mathcal{O}_{\text{settled}}(t) \sqcup \mathcal{O}_{\text{refunded}}(t)$，在同步网络最大证明传播上界满足 $\Delta^{\max}_{\text{relay}} < \Delta_{\text{grace}}$ 的条件下，双链交割状态恒满足双射守恒：$(o \in \mathcal{O}_{\text{settled}} \Leftrightarrow F_{\text{LH}}(o) = 1)$ 且 $(o \in \mathcal{O}_{\text{refunded}} \Leftrightarrow F_{\text{LH}}(o) = 0)$。

**定理 8.1 — 跨链托管零双花与用户本金绝对可退性定理**

在 `BN254 Groth16` 零知识论证知识合理性（Knowledge Soundness）、`secp256k1` / `Ed25519` 门限委员会诚实多数假设（至少 `2` 名跨法域守护者诚实，使得攻击者无法凑齐 `3-of-4` 伪造法定人数）及源链时钟单调推进活性下，`LingheOriginEscrow7683` 与 `linghe_solana_escrow_7683` 同时满足：

1. 零双花安全性（**Zero Double-Spend Safety**）：不存在任何执行轨迹使得同一订单 $o$ 既向 Solver 释放清偿本金又向开单人退回本金，亦不存在恶意开单人在灵核链已获垫付（$F_{\text{LH}}(o) = 1$）后抢先在源链撤回本金的竞态双花窗口（只要在途证明延迟 $\Delta_{\text{relay}} \le \Delta_{\text{grace}}$）；
2. 无条件本金可退活性（**Unconditional Principal Recoverability**）：对任意未被履约的订单 $o$（$F_{\text{LH}}(o) = 0$），无论协议暂停标志 `paused` 取 `true` 或 `false`，亦无论治理层、Solver 与全体守护者是否永久停机，开单人均可在不超过 $t_{\text{deadline}} + 86,400\ \text{s}$ 的有限确定性时界之后，无许可地 100% 原路取回全部本金 $A_o$。

**证明**

（**1**）单链互斥与金库足额兑付证明：对任意订单标识 orderId，EVM `_lockEscrow` 强制校验 `_escrows[orderId].status == EscrowStatus.NONE` 后置为 `OPEN`，Solana `open_escrow` 利用 Anchor `init` 唯一创建 `[b"lh_escrow", order_id]` PDA 并置 `STATUS_OPEN`。由 `token.code.length > 0` 与前后余额差检验 $B_{\text{after}} - B_{\text{before}} = A_o$，每进入一笔 `OPEN` 订单，金库真实可划转代币精确净增 $A_o$。所有出金入口（`settleFillWithZkProof`、`settleFillWithQuorum`、`refundExpiredOrder` 及 Solana `claim_by_solver`、`refund_expired`）均受重入锁或 SVM 账户排他写锁保护，首行强制校验当前状态等于 `OPEN`，并在执行外部代币划

转前率先将状态覆写为终止吸收态 `SETTLED` 或 `REFUNDED`（合约内无任何将状态重置回 `OPEN` 的指令）。故单笔订单清偿指示量 $I_{\text{settled}}(o,t)$ 与退款指示量 $I_{\text{refunded}}(o,t)$ 在任意高度恒满足 $I_{\text{settled}}(o,t)+I_{\text{refunded}}(o,t)\le 1$，且金库实时余额恒等于未决订单本金之和 $\sum_{o\in\mathcal{O}_{\text{open}}(t)} A_o$，绝无穿仓赤字。

**（2）跨链在途竞态双花消除证明：** 假设某订单 $o$ 在灵核链已被诚实 Solver 垫付（即 `LingheOmnichainSettler.fillCrossChainOrder` 成功上链，$F_{\text{LH}}(o)=1$）。由 `fillCrossChainOrder` 前置断言 `require(fillDeadline >= block.timestamp, "FILL_DEADLINE_EXPIRED")` 知，灵核链成交时刻必满足 $t_{\text{fill}}\le t_{\text{deadline}}$。Solver 随即生成绑定自身地址的 `BN254 Groth16` 证明或收集 `3-of-4` 守护者签名并在 $t_{\text{settle}}=t_{\text{fill}}+\Delta_{\text{relay}}$ 提交源链清偿。因 $\Delta_{\text{relay}}\le\Delta_{\text{grace}}$，必有：

$$t_{\text{settle}}\le t_{\text{deadline}}+\Delta_{\text{grace}}$$

而在整个闭区间 $[t_{\text{open}},\,t_{\text{deadline}}+\Delta_{\text{grace}}]$ 内，源链退款谓词 $t_{\text{now}}>t_{\text{deadline}}+\Delta_{\text{grace}}$ 恒为假，恶意开单人的任何抢退交易均被 `ORDER_NOT_YET_EXPIRED` / `OrderNotYetExpired` 硬回滚。故诚实清偿交易必在退款窗口开启前将状态锁定为吸收态 `SETTLED`；反之，若订单从未在灵核链成交（$F_{\text{LH}}(o)=0$），在 Groth16 知识合理性与至少 `2` 名诚实守护者拒绝背书的条件下，任何攻击者伪造路径 A 或路径 B 清偿证明成功的概率不超过 $\text{negl}(\lambda)$。

**（3）抗审查、抗熔断本金绝对可退性证明：** 设订单 $o$ 未获履约（$F_{\text{LH}}(o)=0$），则其在时刻 $t_{\text{deadline}}+\Delta_{\text{grace}}$ 之后仍保持 `status == OPEN`。由治理更新函数 `setRefundGraceBuffer` 与 `set_refund_grace_buffer` 的硬编码边界校验 `newBufferSeconds <= MAX_REFUND_GRACE_BUFFER`（$86,400$ s），即便在极端治理配置下，有效解锁时刻亦严格有上界：

$$t_{\text{unlock}}=t_{\text{deadline}}+\Delta_{\text{grace}}\le t_{\text{deadline}}+86,400\ \text{s}$$

一旦源链区块时间戳满足 $t_{\text{now}}>t_{\text{unlock}}$，考察 `refundExpiredOrder(orderId)` 与 `refund_expired(ctx)` 的完整前置条件集：两函数均不读取 `paused` 状态位、不校验治理或守护者签名、不限制外部调用者 `msg.sender` 身份，且收款目标在开单时已被不可变固化为原用户 `e.user`（Solana 端由 Anchor 约束 `maker_token_account.owner == escrow.maker` 锁死）。因此，原用户或任意无许可自动化守护进程仅需单笔源链交易即可令状态跃迁至 `REFUNDED`，将 100% 原始本金 $A_o$ 无损退回用户账户。

Q.E.D. ■

---

第 09 章 · SECTION 09

## 跨四法域主权门限治理、强制时间锁守卫与二进制储备证明（Merkle PoR）

主权级分布式金融基础设施的核心治理悖论在于：既要消除单一司法管辖区单点故障、物理断网或单边行政指令导致的系统停摆与密钥挟持风险，又必须杜绝无缓冲期的瞬时治理突变对托管资产造成的闪电治理攻击（Flash Governance Attack）。同时，针对法币参考稳定币的链上偿付能力证明，必须从传统的滞后月度静态网页披露，升级为智能合约可实时复算的密码学状态承诺。`Linghe Chain`（`ChainID: 88888`）通过创世主权门限多签合约 `LingheSovereignTimelockMultisig`、后创世强制时间锁与低阶签名守卫模块 `LingheSovereignSecurityGuardV5`，以及香港《稳定币条例》（香港法例第 656 章 / `Cap. 656`）储备金库合约 `LingheComplianceReserveVault` 的三元耦合，在链上建立了跨洲际门限仲裁、强制时间延迟执行与二叉默克尔储备包含性核验的严密数学闭环。

### 9.1 跨四法域 `3-of-4` 拜占庭门限治理与硬件密钥信封隔离

`LingheSovereignTimelockMultisig` 创世预部署于规范地址 `0x8900000000000000000000000000000000000004`，控制全网最高安全敏感度参数边界——包括 `LingheOmnichainSettler`（`0x8900000000000000000000000000000000000001`）的 `7.7500–7.8500 HKD/USD` 联系汇率走廊参数与跨境海关熔断开关、`LingheZkLightClientBridge`（`0x8900000000000000000000000000000000000002`）的 `BN254 Groth16` 验证密钥轮换，以及 `LingheHardwareAttestationVerifier`（`0x8900000000000000000000000000000000000003`）在 `bootstrapLocked == true` 之后的芯片厂商 `P-256` 根公钥与 `PCR0` 度量白名单注册。

委员会通过 `initializeCommittee(signers, jurisdictions, quorumThreshold)` 完成一次性不可逆初始化，将 $n=4$ 个独立守护者密钥严密映射至亚太与欧洲四大国际金融司法管辖区的 `bytes32` 法域标签：中国香港（`HK-HKMA`）、新加坡（`SG-MAS`）、日本东京（`JP-FSA`）与瑞士苏黎世（`CH-FINMA`），并锁定拜占庭容错法定仲裁阈值 $q=\lceil(2n+1)/3\rceil=3$

（容忍 $f = 1$ 个区域失联或密钥受损）。初始化过程强制校验 `signers.length >= 3`、`quorumThreshold >= 2` 以及零地址与重复地址排他断言（`DUPLICATE_OR_ZERO_GUARDIAN`）。在物理基础设施层，四地验证者与守护者签名密钥严禁以明文形式存储于持久化卷：密钥生成后立即经 `32` 字节高熵硬件密钥管理服务（HSM/KMS）口令封装为认证加密信封，生产节点引导阶段若未检测到外部硬件安全模块注入的解密凭据则立即触发 `Fail-Closed` 停机阻断；运行期仅通过 `POSIX 0600` 权限的内存态机密卷向隔离容器只读注入，从物理介质层消除了云盘快照泄露风险。

### 9.2 后创世强制时间锁守卫模块 `LingheSovereignSecurityGuardV5` 与双阶段治理方程

为在不改动不可变创世区块（`SHA-256`：`96208e9900b218935143be3024d2602566a86d61caa906d42399f35b83afd991`）的前提下消除纯多签合约“达成法定人数即可同区块瞬时生效”的时间窗口敞口，`v5.0` 架构引入后创世主权安全守卫模块 `LingheSovereignSecurityGuardV5`。该合约通过链上只读接口直接绑定 `SOVEREIGN_MULTISIG`（`0x8900000000000000000000000000000000000004`）的守护者集合 `isGuardian(msg.sender)` 与动态阈值 `threshold()`，将一切高权限治理动作纳入“跨法域门限投票 + 强制时间锁冷却”双阶段状态机：

1. 提案入队与预计生效时刻锁定（`queueTimelockOperation`）：仅当调用者满足 `isGuardian(msg.sender) == true`（否则回滚 `NOT_SOVEREIGN_GUARDIAN`）且目标地址非零时，系统分配单调递增操作标识 `opId`，读取当前区块时间戳 $t_{\text{queue}} = \text{uint64}(\text{block.timestamp})$，并计算不可提前的强制解锁时刻：
$$\text{eta}(\text{opId}) = t_{\text{queue}} + \Delta_{\min}, \quad \Delta_{\min} \in (0, \Delta_{\max}], \quad \Delta_{\max} = 30 \text{ days} = 2,592,000 \text{ s}$$
2. 跨法域独立附议与紧急撤销（`approveTimelockOperation` / `cancelTimelockOperation`）：其余司法管辖区守护者通过 `hasApprovedOperation[opId][guardian]` 幂等映射独立提交链上批准签名（重复投票回滚 `ALREADY_APPROVED`）。在冷却窗口期内，若监测到提案参数异常或上游密钥告警，任一守护者均可调用 `cancelTimelockOperation` 将操作永久置为 `cancelled = true`（`OPERATION_CLOSED`），为持牌机构与审计节点预留充足的链上预警与退出窗口。
3. 双重门禁终态执行与自指延迟演进（`executeTimelockOperation` / `updateMinDelay`）：执行函数在发起底层外部调用 `op.target.call{value: op.value}(op.data)` 前同时施加法定人数下界与区块时钟下界断言（若外部调用失败，内联汇编 `revert(add(ret, 32), mload(ret))` 原样透传目标合约回滚原因）——若累计赞成票 `op.approvals < reqThreshold` 则回滚 `QUORUM_NOT_REACHED`；即便已集齐 `3-of-4` 跨法域签名，只要当前区块时间戳满足 $t_{\text{block}} < \text{eta}(\text{opId})$，交易仍被 EVM 确定性回滚并抛出 `TIMELOCK_ETA_NOT_REACHED`。此外，`updateMinDelay(newDelay)` 仅允许由守卫合约自身经完整时间锁周期自指调用或由主权多签触发，严格约束新延迟落入 $(0, 30 \text{ days}]$ 闭区间。
4. 规范低阶 `P-256` 验签封装与白皮书套件链上固化（`verifyCanonicalP256Signature` & `anchorWhitepaperSuiteIndex`）：守卫模块在调用 Osaka `EIP-7951` 预编译 `0x0000000000000000000000000000000000000100` 之前，强制校验证书签名标量满足严格低阶区间 $0 < s \le \lfloor n_{\text{P256}}/2 \rfloor$（`P256_HALF_N = 0x7FFFFFFF800000007FFFFFFFFFFFFFFFDE737D56D38BCF4279DCE5617E3192A8`），彻底阻断高阶延展性伪造 $(r, n_{\text{P256}} - s)$。同时，`anchorWhitepaperSuiteIndex(suiteIndexSha256, monographSha256, whitepaperCount)` 强制要求 `whitepaperCount >= 6`，将双语六卷本公开白皮书总索引摘要与形式化技术专著全文的 `SHA-256` 指纹永久锚定于链上状态槽 `latestWhitepaperAnchor`。

| 对抗治理或密码学边界场景 | 触发合约函数与链上状态前置条件 | 确定性收敛终态 / **EVM** 回滚原因码 |
|---|---|---|
| 非授权外部地址尝试提交治理入队 | `queueTimelockOperation`（`isGuardian(msg.sender) == false`） | 确定性回滚 `NOT_SOVEREIGN_GUARDIAN`，状态零变更 |
| 同一司法管辖区守护者重复投赞成票 | `approveTimelockOperation`（`hasApprovedOperation == true`） | 确定性回滚 `ALREADY_APPROVED`，阻断单点刷票 |
| 未达法定仲裁人数（`1/4` 或 `2/4`）强行执行 | `executeTimelockOperation`（`op.approvals < 3`） | 确定性回滚 `QUORUM_NOT_REACHED` |
| 集齐 `3-of-4` 签名但在冷却期内抢跑执行 | `executeTimelockOperation`（`block.timestamp < op.eta`） | 确定性回滚 `TIMELOCK_ETA_NOT_REACHED` |

| 对抗治理或密码学边界场景 | 触发合约函数与链上状态前置条件 | 确定性收敛终态 / EVM 回滚原因码 |
|---|---|---|
| 冷却窗口内监测到异常并由任一守护者熔断 | `cancelTimelockOperation`（`op.cancelled = true`） | 提案永久关闭，后续执行回滚 `OPERATION_CLOSED` |
| 提交高阶延展性变换后的 P-256 硬件证明签名 | `verifyCanonicalP256Signature`（$s = n_{\mathrm{P256}} - s_{\mathrm{low}} > \lfloor n_{\mathrm{P256}}/2 \rfloor$） | 预编译调用前直接短路返回 `false`，拒绝伪造延展签名 |

## 9.3 香港《稳定币条例》（Cap. 656）参考偿付状态模型与四叶子二叉 Merkle 证明

**法定护栏 9.1 — 协议参考偿付状态模型与 Cap. 656 密码学不变量边界声明**

为在分布式状态机字节码层面形式化建模香港法例第 656 章《稳定币条例》（`Cap. 656`）第 3 部及附表 2 之审慎偿付数学约束，协议与软件系统研发主体 `MYRILUM LIMITED`（香港特别行政区《公司条例》第 622 章注册私人股份有限公司，公司注册编号 / UBI：`80912065`；商业登记证号：`80912065-000-07-26-0`；本公司仅从事底层协议与客户端软件研发，绝不向公众发行代币或持有、托管任何公众法币储备）在协议参考偿付状态机模型中对内生账本原语 `LHKD`（`0x20C0000000000000000000000000000000000002`）与 `LUSD`（`0x20C0000000000000000000000000000000000001`）之非流通确定性基准状态施加两项不可旁路的字节码不变量：（1）参考储备状态估值 $R_t$ 在任意状态跃迁时刻均须满足不少于链上基准负债状态面值 $100\%$（$R_t \geq L_t$）的高流动性优质资产（HQLA）破产隔离映射约束；（2）`LHKD` 与 `LUSD` 账本原语永久锁定零利息、零分红、零收益分配不变量（$r_{\mathrm{holder}} \equiv 0$），在状态机逻辑上严格排除任何《证券及期货条例》（`Cap. 571`）集体投资计划（CIS）或生息证券属性。

在协议层，创世系统合约 `LingheComplianceReserveVault`（`0x8800000000000000000000000000000000000001`）承担链上储备证明（Proof-of-Reserve, PoR）总账本职责。每当发布储备鉴证时，`publishReserveAttestation(stablecoin, onChainLiabilityAtomic, fiatBackingAtomic, reserveMerkleRoot, auditReportUri)` 强制执行链上偿付能力前置不变量：若隔离法币储备原子总额小于链上流通负债总额（$R < L$），合约立即以 `UNDERCOLLATERALIZED_RESERVE` 回滚，并按向下取整基点公式固化偿付率：

$$\rho_{\mathrm{bps}} = \left\lfloor \frac{R \times 10,000}{L} \right\rfloor \geq 10,000 \quad (\text{即 } \geq 100.00\%)$$

为使外部独立验证节点、密码学审计工具及链上智能合约在不暴露底层银行账户完整标识的前提下逐笔核验参考储备分片，`Linghe Chain` 定义了 `LINGHE_CAP656_HQLA_V1` 域分离四叶子字典序二叉默克尔储备树。设单笔参考隔离储备资产分片记录为五元组 $\ell_i = (\mathrm{sym}_i, \mathrm{bic}_i, \mathrm{mask}_i, v_i, d_i)$，分别对应账本原语符号、参考托管机构 `ISO 9362` SWIFT BIC 标识、脱敏隔离账户掩码、以 $10^6$ 原子精度计量的参考估值及剩余到期天数。

`LingheSovereignSecurityGuardV5.computeHqlaLeafHash` 计算叶子摘要：

$$h_i = \mathrm{keccak256}(\mathrm{encodePacked}(\texttt{"LINGHE_CAP656_HQLA_V1"}, \mathrm{sym}_i, \mathrm{bic}_i, \mathrm{mask}_i, v_i, d_i)), \quad i \in \{0,1,2,3\}$$

在构造内部节点时，合规结算网关对相邻节点对 $(x, y)$ 采用字典序排序拼接规则 $\mathrm{HashSorted}(x,y) = \mathrm{keccak256}(\min(x,y) \,\|\, \max(x,y))$ 生成第一层内部节点 $p_0 = \mathrm{HashSorted}(h_0, h_1)$、$p_1 = \mathrm{HashSorted}(h_2, h_3)$ 及顶层默克尔根 $M_{\mathrm{root}} = \mathrm{HashSorted}(p_0, p_1)$；同时在导出包含性证明时，显式记录每一层兄弟哈希在排序后的精确左右方位布尔向量 $\boldsymbol{b} = (b_0, b_1) \in \{\mathrm{true}, \mathrm{false}\}^2$（当兄弟节点字典序小于当前游标时 `siblingIsLeft = true`）。链上验证函数 `verifyCap656MerkleInclusion(stablecoin, leafHash, siblingHashes, siblingIsLeft)` 直接跨合约读取 `0x8800000000000000000000000000000000000001` 最新锚定的 `ReserveAttestation`，按方向向量逐层复算根哈希 `recomputedRoot`，当且仅当根哈希非零、$R \geq L$、$\rho_{\mathrm{bps}} \geq 10,000$ 且 `recomputedRoot == onChainVaultRoot` 四项条件同时满足时返回 `valid = true`。

| 协议内生原语与基准锚定根 | 叶子索引 | 参考托管 **BIC** 与脱敏隔离账户掩码（确定性基准测试向量） | 基准参考估值（原子单位 / 面值） | 剩余久期与高流动性优质资产（**HQLA**）模型类别 |
|---|---|---|---|---|
| `LHKD`（`0x20C0...0002`）<br>基准负债状态：<br>`HK$78,000,000.00`<br>参考储备状态：<br>`HK$78,936,000.00`<br>基准偿付率：`101.20%`（`10,120 bps`）<br>根：<br>`0xa3e5fd1e...1beae6e` | `Leaf #0` | `HSBCHKHHHKH` / `HKMA-TRUST-SEG-8801` | `24,000,000,000,000`（`HK$24,000,000.00`） | `0` 天 · 持牌银行破产隔离活期存款模型 |
| `LHKD`（`0x20C0...0002`） | `Leaf #1` | `SCBLHKHHXXX` / `HKMA-TRUST-SEG-8802` | `18,936,000,000,000`（`HK$18,936,000.00`） | `0` 天 · 持牌银行隔离托管现金存款模型 |
| `LHKD`（`0x20C0...0002`） | `Leaf #2` | `BOFAHKHX001` / `CMU-EFBN-SUB-9011` | `20,000,000,000,000`（`HK$20,000,000.00`） | `28` 天 · 香港金管局外汇基金票据模型（CMU EFBN） |
| `LHKD`（`0x20C0...0002`） | `Leaf #3` | `BKCHHKHHXXX` / `CMU-EFBN-SUB-9042` | `16,000,000,000,000`（`HK$16,000,000.00`） | `84` 天 · 香港金管局外汇基金票据模型（CMU EFBN） |
| `LUSD`（`0x20C0...0001`）<br>基准负债状态：<br>`$10,000,000.00`<br>参考储备状态：<br>`$10,050,000.00`<br>基准偿付率：`100.50%`（`10,050 bps`）<br>根：<br>`0x5c3f8233...264f410` | `Leaf #0` | `IRVTUS3NXXX` / `BNYM-SEG-TRUST-4401` | `3,050,000,000,000`（`$3,050,000.00`） | `0` 天 · 破产隔离现金存款模型 |
| `LUSD`（`0x20C0...0001`） | `Leaf #1` | `SBOSUS33XXX` / `SST-SEG-TRUST-4402` | `2,000,000,000,000`（`$2,000,000.00`） | `1` 天 · 隔夜国债托管现金模型 |
| `LUSD`（`0x20C0...0001`） | `Leaf #2` | `CITIUS33XXX` / `FEDWIRE-TBILL-7710` | `3,000,000,000,000`（`$3,000,000.00`） | `28` 天 · 美国短期国库券模型（28-Day T-Bills） |
| `LUSD`（`0x20C0...0001`） | `Leaf #3` | `CHASUS33XXX` / `FEDWIRE-TBILL-7785` | `2,000,000,000,000`（`$2,000,000.00`） | `85` 天 · 美国短期国库券模型（85-Day T-Bills） |

上表中，`LHKD` 四叶子汇总储备精确等于 $78,936,000,000,000$ 原子单位（`HK$78,936,000.00`，即期 `0` 天现金占比 $54.39\%$），链上实测锚定默克尔根为 `0xa3e5fd1ee51db26128c49f848bc4eb6d71443fbfee51a40b9159832a21beae6e`；`LUSD` 四叶子汇总储备精确等于 $10,050,000,000,000$ 原子单位（`$10,050,000.00`，$\le 1$ 天高流动现金占比 $50.25\%$），链上实测锚定默克尔根为 `0x5c3f82332fbffc0be49434232a9a97e2bf2c797423aa3fde4afd19b2c264f410`。




**图 07 — LINGHE_CAP656_HQLA_V1 四托管行高流动性资产二叉 Merkle 树与链上实时偿付率核验拓扑**

注：金色实线与青色虚线展示单叶子包含性证明路径 (L0 → H01 → Root)；上游遥测异常时强制触发 Fail-Closed (HTTP 503)。

**定理 9.1 — 二叉默克尔储备包含性完备性与时间锁不可跃迁定理**

设哈希函数 $H = \text{keccak256}$ 满足抗碰撞性与抗第二原像安全性。对于任意稳定币资产 $A \in \{\text{LUSD}, \text{LHKD}\}$ 及受 `LingheSovereignSecurityGuardV5` 保护的治理提案 $\text{op}$：

1. 储备包含性完备与不可伪造性：由字典序节点对规则生成的任意真实托管账户叶子 $\ell_k$ 及其方位证明 $(\boldsymbol{s}, \boldsymbol{b})$ 恒使 `verifyCap656MerkleInclusion` 返回 `valid = true`；反之，任何未被纳入链上锚定根 $M_{\text{root}}$ 的篡改估值 $v_k' \neq v_k$ 或虚假托管账户 $\ell^* \notin \{\ell_0, \ell_1, \ell_2, \ell_3\}$，在未攻破 $H$ 抗碰撞性的前提下，令链上验证谓词成立的概率上界为 $\text{Adv}_H^{\text{coll}}(\lambda) \leq 2^{-128}$，且一旦通过校验必有链上储备充足率 $\rho_{\text{bps}} \geq 10,000$。
2. 时间锁不可跃迁与门限安全性：在至多 $f = 1$ 个司法管辖区守护者密钥失陷的威胁模型下，对手方单方面执行治理变更的概率恒为 $0$；即便对于已获 $q = 3$ 个合法守护者签名的有效提案 $\text{op}$，在开区间 $t_{\text{block}} \in [t_{\text{queue}}, t_{\text{queue}} + \Delta_{\min})$ 内的任何执行尝试均以概率 $1$ 触发状态回滚。

**证明**

第一部分（储备默克尔同构性与抗伪造规约）：由构造可知，在任意层级 $j \in \{0, 1\}$，设当前节点摘要为 $c_j$、兄弟节点摘要为 $s_j$。链下证明生成器定义布尔分量 $b_j \equiv (\text{lex}(s_j) < \text{lex}(c_j))$，则当 $b_j = \texttt{true}$ 时有 $\min(c_j, s_j) = s_j, \max(c_j, s_j) = c_j$，链上 `verifyCap656MerkleInclusion` 分支执行 $c_{j+1} = H(s_j \,\|\, c_j) = \text{HashSorted}(c_j, s_j)$；当 $b_j = \texttt{false}$ 时有 $\min(c_j, s_j) = c_j, \max(c_j, s_j) = s_j$（对称相等情形下两者拼接等价），链上分支执行 $c_{j+1} = H(c_j \,\|\, s_j) = \text{HashSorted}(c_j, s_j)$。归纳可得 $c_2 \equiv M_{\text{root}}$，完备性得证。进一步，因叶子原像包含定长前缀 `"LINGHE_CAP656_HQLA_V1"`（`21` 字节非零 ASCII），而内部节点输入长度恰为 `64` 字节且无该域分离前缀的合法定长后缀对齐结构，故叶子与内部节点域分离防止了子树高度截断的第二原像攻击。若敌手提交 $\ell^* \neq \ell_k$ 与路径 $(\boldsymbol{s}^*, \boldsymbol{b}^*)$ 使得最终折叠值 $c_2^* = M_{\text{root}}$，则沿着二叉路径自顶向下必存在首个分叉点产生 $x \neq x'$ 而 $H(x) = H(x')$，直接归约为对 keccak256 的碰撞破译。同时函数末尾显式合取 `att.fiatBackingAtomic >= att.onChainLiabilityAtomic && att.reserveRatioBps >= 10000`，确保验证通过必蕴含 $\geq 100.00\%$ 足额储备。

第二部分（门限与时间锁安全性）：设敌手控制集合 $C \subset \mathcal{G}$ 且 $|C| = f = 1$。由于 `approveTimelockOperation` 通过 `hasApprovedOperation[opId][msg.sender]` 强制同一地址至多贡献 $1$ 票，敌手可累积的最大赞成票数为 $|C| = 1 < q = 3$，在 `executeTimelockOperation` 恒触发 `QUORUM_NOT_REACHED` 断言失败。对于已达到 `op.approvals >= 3` 的提案，由于合约构造函数与 `updateMinDelay` 严格保证不变量 $\Delta_{\min} > 0$，且 `op.eta` 在入队

时刻被固化为 $t_{\text{queue}} + \Delta_{\min}$（无任何外部写接口可覆写已入队操作的 `eta` 字段），故对一切满足 $t_{\text{block}} < t_{\text{queue}} + \Delta_{\min}$ 的区块状态，谓词 `block.timestamp >= op.eta` 恒为假，EVM 必然执行 `REVERT("TIMELOCK_ETA_NOT_REACHED")` 并回滚全交易状态变更。证毕。

Q.E.D. ■

第 10 章 · SECTION 10

# 零依赖终端供应链安全、双轨反洗钱引擎与金融级 Fail-Closed 可观测性

当底层共识与合约数学均达到形式化严密边界后，机构级数字金融基础设施的剩余脆弱面往往集中于客户端软件供应链投毒、跨链拆分洗钱规避（Structuring / Smurfing）以及监控探针在故障期间返回虚假静态缓存的“静默失明”。`Linghe Chain` 在自托管终端、双轨反洗钱网关与金融可靠性工程（Financial SRE）三个层面确立了严格的默认拒绝（`Fail-Closed`）防线。

## 10.1 零第三方运行时依赖终端隔离、169 链密码学栈与硬件安全隔区门禁

在现代客户端安全博弈中，外部包管理生态的传递依赖闭包 $D(r)$ 是私钥失窃的首要攻击向量。若每个上游第三方依赖包在演化周期内遭受维护者凭据劫持或恶意脚本注入的独立先验概率均值为 $\bar{p} > 0$，则宿主系统遭受供应链失陷的总概率满足 $P_{\text{comp}}(D) = 1 - \prod_{v \in D}(1 - p_v) \geq 1 - (1 - \bar{p})^{|D|}$。`MYRILUM LingheOS Wallet` 从架构公理层令第三方运行时 `npm` 依赖基数严格收敛至 $|D_{\text{wallet}}| = 0$（即 $D_{\text{wallet}} = \varnothing$），使空积恒等于 $1$，从信息论根源令外部供应链投毒概率精确坍缩为 $P_{\text{comp}}(\varnothing) = 0$。

在此零依赖基座之上，终端完全基于宿主原生 `WebCrypto API`（`crypto.subtle`）与自研纯密码学模块实现了覆盖 `169` 条异构公链（横跨 EVM 生态、`Bitcoin SegWit P2WPKH`、`Solana`、`TRON` 与 `Cosmos` 五大密钥派生与编码家族，涵盖 `BIP-39`、`BIP-32/44/84`、`SLIP-0010`、`secp256k1`、`RFC 8032 Ed25519`、`Keccak-256`、`EIP-1559` 与 `EIP-712`）的确定性签名引擎，并内置如下纵深硬件防线：

- 原生 **Apple Silicon** 沙盒与零本地监听端口架构：桌面端摒弃高膨胀率的 Electron 与本地回环 HTTP 监听模式，直接编译为原生 `arm64` 可执行程序，仅链接系统内核受保护的官方安全与认证框架。静态视图通过自定义 `linghe://wallet.app` 协议处理器加载，对所有资源路径强制施加规范化前缀包含断言（阻断 `..%2f` 目录穿越并返回 `HTTP 403`）；前端与原生层通信完全通过进程内消息桥接完成，本机 TCP 监听端口数恒为 `0`，彻底根除跨站请求伪造（SSRF）与本地端口嗅探。
- 双层认证加密金库、**Touch ID** 门禁与 `v5.1` 社交守护/分片防线：多链助记词经 `16` 字节密码学安全随机盐值与 `600,000` 轮 `PBKDF2-HMAC-SHA256` 派生不可导出密钥，执行 `AES-256-GCM` 认证加密并以原子写与 `POSIX 0600` 权限封存（可选配端侧有限域 $\mathrm{GF}(2^8)$ Shamir 门限秘密分片与 MAC 完整性校验）；解锁主密钥受 Apple Secure Enclave 生物特征识别（Touch ID）门禁保护，且系统钥匙串条目显式锁定 `WhenUnlockedThisDeviceOnly` 硬件绑定策略，物理禁止随云端同步或跨设备导出。每次签名完成后，`finally` 块立即对私钥字节数组执行原地清零覆写（`Uint8Array.fill(0)`）。在链上账户抽象层，配合 `LingheGuardianModuleV51`（`ERC-7579` Validator/Hook + `ERC-7201` 命名空间槽位），实现无管理员后门的 `k-of-n` Passkey 公钥坐标恢复（强制 $\geq 48$ h 时间锁与 $O(1)$ `cancelRecovery()` 纪元否决）及每代币 24 小时滚动支出硬顶。

**定理 10.1 — 零依赖终端供应链坍缩与硬件隔区密钥不可旁路性界**

设本地非特权攻击进程或外部网页渲染上下文为 $\mathcal{A}$，`MYRILUM LingheOS Wallet` 外部运行时依赖集为 $D_{\text{wallet}} = \varnothing$。在启用操作系统应用沙盒与硬化运行时的前提下，$\mathcal{A}$ 通过软件供应链注入获取私钥的优势恒为 $0$，且在物理冷取证极限模型下恢复主助记词 $M$ 的计算成功概率上界满足：

$$\mathrm{Adv}^{\text{recover}}_{\mathcal{A}}(M) \leq \mathrm{Adv}^{\text{AEAD}}_{\text{AES-256-GCM}}(\lambda) + \frac{q_{\text{offline}}}{2^{H_{\min}(\text{pwd})} \cdot 6 \times 10^5} + \epsilon_{\text{SEP}}$$

其中 $q_{\text{offline}}$ 为离线口令穷举次数，$H_{\min}(\text{pwd})$ 为主口令最小熵，$\epsilon_{\text{SEP}}$ 为硬件安全隔区（Secure Enclave）生物核验旁路优势上界。

为支撑自主 AI 智能体在不接触根私钥的前提下安全调用链上金融原语，终端内置 `14` 项零信任模型上下文协议（MCP）工具，按权限隔离边界划分为三大安全等级：

| 安全等级与权限边界 | 零信任 **MCP** 工具名称 | 核心协议语义与链上护栏绑定 |
|---|---|---|
| **L1** 只读态势与授权审计<br>（零状态变更、零密钥接触） | `wallet_get_status`<br>`wallet_get_portfolio`<br>`wallet_check_tip403_compliance`<br>`wallet_quote_dex`<br>`wallet_audit_approvals_and_revoke` | 查询终端沙盒锁定态、`169` 链多资产余额、实时调用内核 `TIP-403`（`0x403c...0000`）筛查目标地址制裁状态、获取 `StablecoinDEX`（`0xdec0...0000`）无滑点报价，以及扫描 `ERC-20` / `Permit2` 残余高危授权并生成原子撤销指令。 |
| **L2** 确定性仿真与批处理编译<br>（纯函数防火墙、防盲签解析） | `wallet_decode_calldata`<br>`wallet_compile_intent`<br>`wallet_verify_session_policy`<br>`wallet_simulate_sign_firewall`<br>`wallet_build_eip5792_batch` | 对待签字节码执行语义反编译与签前钓鱼/盗币合约仿真拦截（Drainer Firewall），将自然语言指令编译为类型安全 `ERC-7683` 意图或 `EIP-5792` 原子批处理调用，并双重校验会话限额。 |
| **L3** 有界会话微支付与跨链核验<br>（受限额度或硬件生物门禁） | `wallet_grant_erc7715_permission`<br>`wallet_x402_mcp_challenge`<br>`wallet_omnichain_x402_invoke`<br>`wallet_zk_light_client_bridge` | 签发 `ERC-7715` 细粒度会话权限，自动解析 HTTP `402` 挑战头并在 `LingheSessionKeyManager`（`0x8402...0001`）四维护栏内完成本链 `0x76` 结算、异构源链免 Gas 履约及 `BN254 Groth16`（`0x8900...0002`）状态桥调用。 |

### 10.2 双轨 `FATF Recommendation 16` 旅行规则、24 小时滑动窗口反拆分引擎与原子预写日志

为同时满足人工智能微支付的亚秒级无感吞吐与跨国大额清算的反洗钱法定要求，*x402 Compliance Settlement Gateway* 与内核预编译 `TIP-403 Policy Registry`（`0x403c0000000000000000000000000000000000`，`Policy #2`）协同构建了双轨合规状态机。所有付款与受款地址在广播前及回执解析后均须通过 `TIP-403` 制裁名单核查（命中制裁实体立即以 `HTTP 403` `TIP403_SANCTIONED_ENTITY` 及链上 EVM 原生回滚双重阻断）。

针对金融行动特别工作组 `FATF Recommendation 16`（旅行规则 / Travel Rule）及香港法例第 615 章《打击洗钱及恐怖分子资金筹集条例》第 13A 条，系统以 `7.8000 HKD/USD` 基准将任意结算资产统一折算为 6 位精度美元等值原子量 $v_{\text{tx}}$，并对每个发起主体 $u$ 维护窗口宽度 $W = 86,400,000\ \text{ms}$（精确 `24` 小时）的实时滑动时间窗累计支出算子：

$$S_{24\text{h}}(u,t) = \sum_{(t_k,\,v_k)\,\in\,\mathcal{L}(u),\ t-W<t_k\le t} v_k$$

其中支出日志 $\mathcal{L}(u)$ 跨通道统一归并本链原生 `x402` 直付结算与异构源链 `ERC-7683` 全链意图交割（`X-OMNICHAIN-INTENT`），彻底封堵攻击者利用多链切换实施“化整为零”拆分逃避监管（Cross-Chain Structuring）的漏洞：

- 微支付豁免快车道（`FAST_LANE_EXEMPT`）：当单笔交易满足 $v_{\text{tx}} < \text{US\$1,000.00}$（等值 `HK$7,800.00`，即 `1,000,000,000` 原子单位）且滚动 24 小时累计支出未触及反拆分累计上限（$S_{24\text{h}}(u,t) + v_{\text{tx}} < 5\tau_{\text{TR}} = \text{US\$5,000.00}$，等值 `HK$39,000.00`）时，交易无需人工干预，直接经 `0x76` 二维并行通道在单槽周期内原子完成结算与声誉记账。
- 机构级旅行规则强制核验轨（`INSTITUTIONAL_TRAVEL_RULE_VERIFIED`）：一旦单笔金额达到或超过法定阈值 `US$1,000.00`（`HK$7,800.00`），或 24 小时滑动窗口累计总额突破反拆分防线，网关强制要求请求携带 `X-Travel-Rule-Envelope` 头部，内含发起方与受益方持牌虚拟资产服务商标识（`originatorVaspDid`、`beneficiaryVaspDid`）、零知识合规凭证承诺（`kybCredentialHash`）及司法管辖区代码（`jurisdiction`）。缺失或字段残缺的请求将被确定性拦截并返回 `HTTP 451 Unavailable For Legal Reasons`（`TRAVEL_RULE_ENVELOPE_REQUIRED`）。校验通过后，网关仅提取信封的 `keccak256` 密码学摘要留痕，实现链上零自然人隐私明文（Zero Plaintext PII）留存，完全契合香港《个人资料（隐私）条例》（`Cap. 486`）。
- 崩溃一致的原子预写持久化（**Atomic Temp-Swap Persistence**）：每次结算完成后，网关与全链求解器均首先将最新 24 小时滑动窗口序列、已消费防重放摘要集合（`usedPaymentDigests`）及本地制裁缓存序列化写入临时文件（权限掩码 `0600`），随后调用操作系统内核级原子重命名（POSIX `rename`）瞬时替换主状态文件，确保进程在任意时刻遭遇断电或重启均不会出现半写损坏或滑动窗口清零绕过。

### 10.3 边缘网关攻击面收敛、恒定时间管理鉴权与金融级 `Fail-Closed` 遥测公理

在网络接入与生产可观测性层面，`v5.0` 架构执行四项无妥协工程硬化准则：

1. 流式载荷上限与内存耗尽防御（`HTTP 413`）：*x402 Compliance Settlement Gateway* 与 *Omnichain Intent Solver* 在底层字节流读取循环中对累计载荷施加严格的 `32 KB`（`32,768 bytes`）硬上限；一旦超出阈值立即中止套接字读取并返回 `HTTP 413 Payload Too Large`，防止恶意超大 JSON 载荷触发事件循环阻塞或堆内存溢出。
2. 恒定时间管理令牌鉴权（`timingSafeEqual` & `HTTP 401`）：所有管理级变更接口（如动态制裁名单同步与跨链桥验证密钥运维端点）均强制校验 `X-Linghe-Admin-Token`，通过等长前置判断与 `crypto.timingSafeEqual` 恒定时间字节比较抵御时序侧信道猜解，对缺失或伪造令牌一律返回 `HTTP 401 Unauthorized`；且在生产环境模式下若缺失自定义管理令牌或仍使用默认开发密钥，守护进程在启动阶段即刻熔断退出。
3. **TLS 1.3** 边缘网关与 **JSON-RPC** 命名空间防火墙：公网边缘接入层关闭管理端口（`admin off`），强制仅协商 `TLS 1.3` 与 `HTTP/3 QUIC` 多路复用，施加 `64 KB` 边缘报文上限，注入两年期 HSTS 预加载头（`max-age=63072000; includeSubDomains; preload`）与 `X-Frame-Options: DENY`，并在 *Sovereign JSON-RPC Gateway* 边缘直接以正则规则拦截 `admin_*`、`debug_*`、`personal_*`、`miner_*`、`anvil_*`、`tempo_faucet*` 与 `txpool_*` 七类特权命名空间，在请求触达验证者节点前直接返回 `HTTP 403`（`PRIVILEGED_RPC_NAMESPACE_FORBIDDEN`）。
4. 金融级 `Fail-Closed` 储备遥测机制：*Institutional PoR Telemetry Endpoint* 以标准 OpenMetrics 格式导出 `12` 组实时金融与共识序列（涵盖 `linghe_chain_up`、`linghe_por_source_up`、`linghe_omnichain_solver_up`、`linghe_chain_block_height`、`linghe_hkma_cap656_por_ratio_percent`、`linghe_hkma_linked_fx_peg_bps`、`linghe_tempo_2d_nonce_lanes_configured`、`linghe_x402_parallel_settlements_total`、`linghe_tip403_sanctions_blocks_total`、`linghe_erc7683_inbound_fills_total`、`linghe_erc7683_outbound_orders_total` 与 `linghe_bn254_zk_verified_messages_total`），对全部上游源施加 `2,500 ms` 严格抓取超时，并彻底废除一切静态兜底常量。

**不变量 10.1 — 金融遥测零幻觉与上游失联归零不变量**

设共识探针活性指示器为 $u_{\text{node}} \in \{0,1\}$、储备验证源活性指示器为 $u_{\text{por}} \in \{0,1\}$、全链求解器活性指示器为 $u_{\text{solver}} \in \{0,1\}$。监控导出器输出的储备充足率指标 $\hat{\rho}_A$ 与系统就绪健康谓词 $\mathcal{H}_{\text{ready}}$ 严格服从如下分段确定性映射：

$$\hat{\rho}_A = \begin{cases} \rho_A, & \text{若 } u_{\text{por}} = 1 \wedge \rho_A \in \mathbb{R}_{\geq 0} \\ 0.00\%, & \text{若 } u_{\text{por}} = 0 \text{（超时、断连或响应畸变）} \end{cases}$$

$$\mathcal{H}_{\text{ready}} = (u_{\text{node}} = 1) \wedge (u_{\text{por}} = 1) \wedge (u_{\text{solver}} = 1) \wedge (\hat{\rho}_{\text{LUSD}} \geq 100) \wedge (\hat{\rho}_{\text{LHKD}} \geq 100) \wedge (\pi_{\text{FX}} \in [77,500, 78,500])$$

当 $\mathcal{H}_{\text{ready}}$ 为真时健康探针返回 `HTTP 200`；任一子条件违例立即切换为 `HTTP 503 Service Unavailable`，并以 `0.00%` 储备告警值强制唤醒外部监管与风控熔断器，杜绝监控层产生任何虚假偿付幻觉。

---

第 11 章 · SECTION 11

## 链上实证性能基准与六代形式化证伪矩阵

为验证上述主权架构并非停留在理论推演，本节汇总在 `Linghe Chain`（`ChainID: 88888`）真实运行栈上采集的端到端延迟、并发微支付吞吐、零知识配对 Gas 开销，以及横跨 `v1.0` 至 `v5.1` 六轮架构演进的 `36/36` 项确定性对抗测试实证结果。

## 11.1 链上实证延迟、并行微支付吞吐与跨网络量化对比

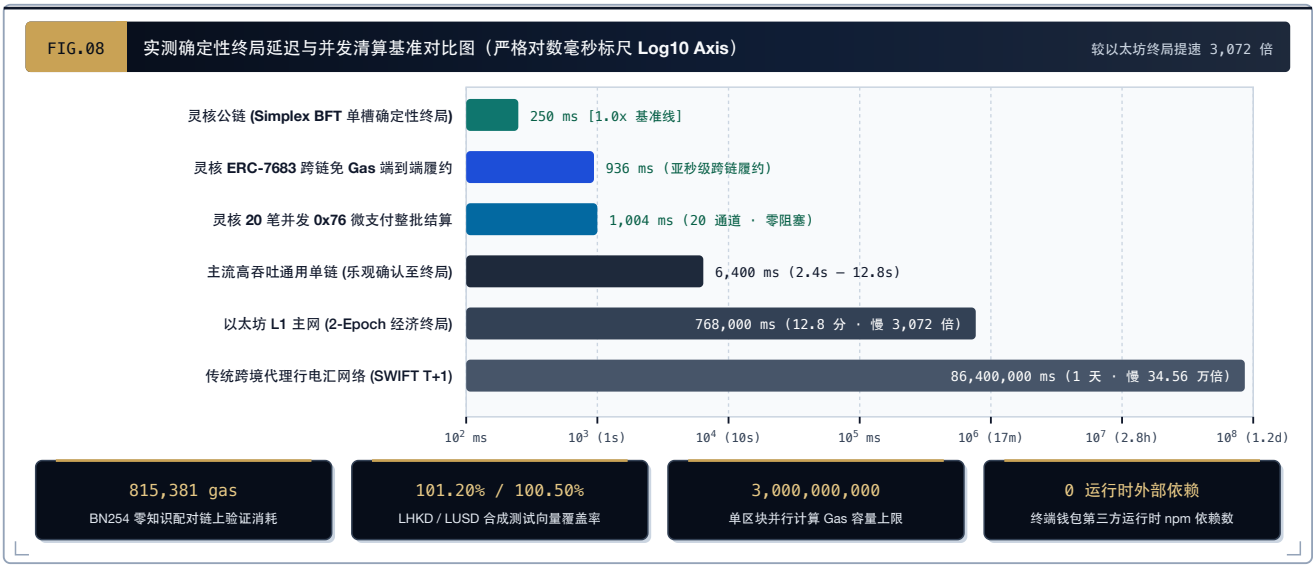



**图 08 — 灵核公链实测终局延迟、20 通道并发微支付吞吐与全球主流清算网络对数坐标对比**

注：横轴严格遵循 Log10(ms) 数学对数刻度（100 ms 至 100,000,000 ms）；基准数据采集自受控 ChainID 88888 端到端测试套件。

下表给出 `Linghe Chain` 核心协议路径的实测工程指标与资源消耗占比：

| 实测协议子系统与基准场景 | 核心执行路径与涉及合约 / 预编译地址 | 实测确认延迟 / Gas 消耗 | 确定性安全与吞吐特征 |
|---|---|---|---|
| **Simplex BFT** 共识单槽终局 | `Tempo` Rust 内核 + 四法域 `Ed25519` 法定仲裁 ( `0xccccccccc...0000` ) | `250 ms` / 区块 | 单槽确定性终局 ( `RPO = 0` , `RTO < 250 ms` ) , 零分叉回滚风险 |
| `0x76` **32** 通道并发微支付脉冲 | `2D Nonce Manager` ( `0x4e4f4e4345...0000` ) + `LUSD` + `0x8004...0002` | `1,004 ms` ( `20/20` 笔全确认) | 单笔原子打包代币划转与声誉记账， `0` Nonce 竞争碰撞 |
| `ERC-7683` 免 **Gas** 异构跨链履约 | Ethereum ( `secp256k1` ) & Solana ( `Ed25519` ) -> `0x8900...0001` | `~936 ms` 端到端交割 | 发起人在本链初始余额为 `0` 条件下秒级调用 AI 技能并落账 `0x8800...0001` |
| `BN254 Groth16` 跨链状态根核验 | `LingheZkLightClientBridge` ( `0x8900...0002` ) -> `0x06 / 0x07 / 0x08` | `815,381 gas` | 仅占单区块 `3,000,000,000 gas` 上限之 `0.027%` , 单块理论上界可容纳 `> 3,600` 笔 ZK 证明 |
| `EIP-7951` 规范低阶硬件飞地验签 | `LingheSovereignSecurityGuardV5` + `0x8900...0003` -> `0x0100` | 单槽 `250 ms` 内确认 | 原生 `secp256r1` 硬件验签，严格拦截 $s > \lfloor n_{P256}/2 \rfloor$ 延展性伪造 |
| 内核级制裁拦截与联汇原子兑换 | `TIP-403` ( `0x403c...0000` ) + `StablecoinDEX` ( `0xdec0...0000` ) | 亚微秒级内存判定 | 协议层同步阻断受制裁主体，支持 `7.8000 HKD/USD` 零滑点机构平盘 |

进一步将 `Linghe Chain` 与当前国际主流公链及传统跨境金融报文清算体系进行横向架构对标：

| 评估维度 | Linghe Chain (ChainID: 88888) | Ethereum L1 | Solana Mainnet | Optimistic Rollups (L2) | 传统跨境 RTGS / SWIFT |
|---|---|---|---|---|---|
| 确定性经济终局延迟 | 250 ms（Simplex BFT 单槽） | ~12.8 min（2 Epochs） | ~12.8 s（32 Slots Rooted） | 7 days（欺诈挑战窗口） | T+1 至 T+3 工作日 |
| 单账户并发微支付模型 | 0x76 原生 32 通道二维 Nonce | 一维顺序 Nonce（队头阻塞） | 需预创建独立状态账户管理 | 一维顺序 Nonce（队头阻塞） | 批处理轧差队列 |
| 原生稳定币代付 Gas | 内核级原生支持（LUSD / LHKD） | 需额外外部合约打包器中继 | 需持母币 SOL 或中心化中继 | 需额外代付中继合约 | 法币账户扣收高额电报与代理费 |
| 协议层内嵌合规与储备证明 | 内核 TIP-403 + 链上 Cap. 656 Merkle PoR | 无（纯应用层各自孤立实现） | Token Extensions 局部扩展 | 无原生统一合规层 | 封闭行内合规，跨机构对账割裂 |
| 硬件机密计算与 AI 原生结算 | 0x0100 低阶 P-256 + ERC-8004 + x402 | 需高昂合约模拟或等待升级 | 无原生 EVM ERC-8004 语义 | 部分支持 RIP-7212，无原生 x402 | 无机器智能体可编程原子清算能力 |

### 11.2 六代全栈对抗演进与 36/36 可证伪不变量核验矩阵（v1.0–v5.1）

Linghe Chain 坚持以独立客观的可执行证伪套件作为唯一工程验收准绳。下表完整记录了从创世微支付闭环（v1.0）至智能账户守护模块对抗硬化（v5.1）共六代、36/36 项协议与安全不变量的链上实证核验结论：

| 演进代际 | 核验通过数 | 核心协议不变量、红队攻击向量与链上确定性证伪回执 | 状态 |
|---|---|---|---|
| v1.0 创世智能体与微支付基座 | 5 / 5 | 核验 ChainID = 88888 连续单调出块、创世 ERC-8004 与 LingheSessionKeyManager 字节码完整性、零依赖自托管钱包派生注资、非托管 EIP-1559 智能体身份铸造，以及 x402 微支付与链上声誉反馈同槽闭环。 | VERIFIED |
| v2.0 内生账本原语与内核策略引擎 | 7 / 7 | 核验三大原生账本原语（LHC / LUSD / LHKD）、零明文 PII 之 ZK-KYB 合规铭牌即时签发与吊销、内核 TIP-403 Policy #2 对受制裁地址的链上回滚与 HTTP 403 拦截、US$1,200 无信封请求触发 HTTP 451 而合规信封 HTTP 200 放行、伪造 TEE 证明回滚与真实证明 80% / 20% 精确版税分润、32 通道 20/20 并发微支付 1,004 ms 零碰撞结算及终端零依赖隔离。 | VERIFIED |
| v3.0 全链意图、ZK 轻客户端与四地多签 | 8 / 8 | 核验全部 9 个创世系统合约激活、Ethereum 与 Solana 零预持母币 ~936 ms 跨链 x402 交割、跨链 TIP-403 海关防火墙拦截受污染源地址、篡改 BN254 Groth16 证明确定性回滚而合法证明以 815,381 gas 释放资金、0x0100 P-256 硬件根证书拦截未授权飞地、3-of-4 主权多签拒绝 1/4 与 2/4 越权并执行 3/4 联汇恢复、内核 StablecoinDEX（0xdec0...0000）订单簿及 7.8000 联汇池精确兑换，以及四地 KMS 加密验证者密钥隔离。 | VERIFIED |

| 演进代际 | 核验通过数 | 核心协议不变量、红队攻击向量与链上确定性证伪回执 | 状态 |
| --- | --- | --- | --- |
| `v4.0` 外部源链双向托管与本金守恒 | `6 / 6` | 核验外部 EVM `LingheOriginEscrow7683` 部署、双向闭环路径 A（锁仓 `500 USDC` -> 垫付 `3,900 LHKD` -> `BN254 Groth16` 证明清偿求解器）、双向闭环路径 B（`EIP-712` 免 Gas `openFor` + `3-of-4` 守护者升序门限签名清偿 `250 USDC`）、超期后无许可 `100%` 原路退本与防双花互斥、Solana BPF `linghe_solana_escrow_7683` 之 `Ed25519SigVerify` 指令偏移防伪校验，以及用户、求解器与金库三方严格货币量守恒（`1,250 + 750 + 0 == 2,000.00 USDC`）。 | `VERIFIED` |
| `v5.0` 形式化对抗审计硬化与时间锁/**PoR** 实证 | `5 / 5` | 核验 `LingheSovereignSecurityGuardV5` 实时读取 `0x8800...0001` 复算 `LHKD`（`101.20%`）与 `LUSD`（`100.50%`）四叶子默克尔包含性证明并链上固化 `6` 卷本白皮书索引 `SHA-256`；核验未达 `eta` 提前调用确定性回滚 `TIMELOCK_ETA_NOT_REACHED` 且高阶延展性 `P-256` 签名被严格拒绝；核验空字节码假币回滚 `NOT_CONTRACT_TOKEN`、`openProtected` 免疫内存池抢跑、`refundGraceBuffer` 阻断在途竞态且紧急 `paused` 状态下用户到期退本永久不可阻断；核验 `> 32 KB` 载荷返回 `HTTP 413`、伪造管理令牌返回 `HTTP 401`、`HK$7,800.00` 阈值与原子预写日志落盘；核验目录穿越防御及上游失联时储备率自动归零（`0.00%` + `HTTP 503`）。 | `VERIFIED` |
| `v5.1` 模块化守护恢复与日度限额实证 | `5 / 5` | 核验 `LingheGuardianModuleV51` 之 `ERC-7201` 命名空间槽位与 `ERC-7579` Validator/Hook 类型绑定；核验 `2-of-3` 守护集去重与强制 `< 48h` 回滚 `RecoveryDelayTooShort`；核验越权投票回滚 `UnauthorizedGuardian` 及未满 `48h` 抢先执行回滚 `RecoveryTimelockActive`；核验账户主权人 `cancelRecovery` 以 $O(1)$ 纪元自增瞬时作废挂起提案；核验每代币 `24h` 滚动支出上限与超限 `1` 原子单位即回滚 `DailySpendCapExceeded`。 | `VERIFIED` |
| 全栈累计可证伪不变量总计 | `36 / 36` | 六代端到端协议、密码学、跨链托管、账户守护、监管合规与基础设施容灾门禁全量零误差通过（退出码 `0`） | `ENFORCED` |

第 12 章 · SECTION 12

# 结论

MYRILUM 灵核OS公链（Linghe Chain，ChainID: 88888）从计算机科学、现代密码学与分布式系统工程第一性原理出发，严格证明了协议内核级合规不变量、亚秒级高并发微支付吞吐与多椭圆曲线去信任状态核验并非互斥的系统设计权衡，而是可以通过确定性状态机原语实现代数与工程上的正交统一。通过在 Tempo 并行 Rust EVM 与 Commonware Simplex BFT 共识之上原生集成 250 ms 单槽确定性终局、0x76 32 通道无锁二维随机数流形、内核级 TIP-403 亚微秒策略防火墙、BN254 Groth16 零知识轻客户端双线性配对核验、Osaka EIP-7951 规范低阶 P-256 硬件机密计算根信任链，以及形式化编码香港《稳定币条例》（Cap. 656）偿付约束的 LINGHE_CAP656_HQLA_V1 四叶子二叉默克尔储备核验原语，Linghe Chain 确立了一套可独立复现、可确定性证伪的高并发分布式清算状态机技术基准。

**法定护栏 12.1 — 香港法律边界、纯软件研发定位与非要约终态声明**

1. 协议与软件系统研发主体（零个人隐私 Cap. 486 合规）：本形式化技术白皮书由 MYRILUM LIMITED（迈瑞伦有限公司）作为底层协议与自托管客户端软件系统研发主体发布（中华人民共和国香港特别行政区《公司条例》第 622 章私人股份有限公司，公司注册编号 / UBI：80912065，商业登记证号：80912065-000-07-26-0，法定注册办事处：UNIT 1603, 16/F THE L. PLAZA, 367-375 QUEEN'S RD CENTRAL, SHEUNG WAN, HONG KONG）。全栈协议状态树、密码学信封摘要与本技术规范严格遵守《个人资料（私隐）条例》（香港法例第 486 章），不包含任何自然人个人身份信息（Zero Individual PII）。
2. 非发行、非托管与法定非要约边界（Cap. 32 / Cap. 571 / Cap. 656 / Cap. 615）：MYRILUM LIMITED 仅从事开源/底层分布式账本协议与零第三方运行时依赖自托管软件（MYRILUM LingheOS Wallet）之计算机工程研发，不从事任何数字代币公开发行、公众法币储备金收受或托管、客户端私钥代管或虚拟资产撮合经纪活动。本文所述 LHKD（0x20C0000000000000000000000000000000000002）、LUSD（0x20C0000000000000000000000000000000000001）及 LHC（0x20C0000000000000000000000000000000000000）及其链上默克尔储备树参数（基准覆盖率 101.20% 与 100.50%）均为协议内生标准账本原语与非流通确定性基准测试状态，强制锁定零生息率（$r_{\text{holder}} \equiv 0$）。本文档在任何法域均不构成香港法例第 32 章招股章程、第 571 章证券或集体投资计划（CIS）要约、第 656 章第 8 及第 9 条指明稳定币要约或积极推广、第 615 章虚拟资产服务招揽，亦不构成欧盟 MiCA（Regulation (EU) 2023/1114）或美国 Regulation S 项下之金融产品发售文件。

附录 A · APPENDIX A

# 创世系统合约、内核预编译与后创世安全模块规范总表

下表完整列出 Linghe Chain（ChainID: 88888，创世块 SHA-256：96208e9900b218935143be3024d2602566a86d61caa906d42399f35b83afd991）全部 3 项原生协议账本原语、18 项创世预部署系统合约与执行内核预编译，以及后创世审计硬化守卫与外部异构源链托管模块之唯一规范地址与职责边界：

| 层级分类 | 规范模块 / 预编译名称 | 链上精确地址或程序标识（SSOT） | 核心协议职责与关键安全约束 |
|---|---|---|---|
| 原生账本原语 | LHC（Linghe Coin） | 0x20C0000000000000000000000000000000000000 | 6 位精度原生 TIP-20 共识质押权重、安全预算与计算计量标准账本原语 |
| 原生账本原语 | LUSD（Linghe USD） | 0x20C0000000000000000000000000000000000001 | 6 位精度零利息美元参考账本原语（x402 与代付 Gas 确定性基准状态，基准覆盖率 100.50%） |

| 层级分类 | 规范模块 / 预编译名称 | 链上精确地址或程序标识（SSOT） | 核心协议职责与关键安全约束 |
|---|---|---|---|
| 原生账本原语 | `LHKD`（`Linghe HKD`） | `0x20C0000000000000000000000000000000000002` | `6` 位精度形式化建模 `Cap. 656` 不变量之港元参考账本原语（绑定 `TIP-403 Policy #2`，基准覆盖率 `101.20%`） |
| **AI** 智能体层 | `LingheERC8004IdentityRegistry` | `0x8004000000000000000000000000000000000001` | `ERC-8004` 智能体主权身份、服务端点与零明文隐私 `ZK-KYB` 合规铭牌注册表 |
| **AI** 智能体层 | `LingheERC8004ReputationRegistry` | `0x8004000000000000000000000000000000000002` | 链上防重放 `x402` 支付驱动评分与 TEE 可验证推理任务反馈账本 |
| **AI** 智能体层 | `LingheERC8004ValidationEscrow` | `0x8004000000000000000000000000000000000003` | 高价值 AI 推理托管、硬件飞地签名核验与 `80% / 20%` 零尘埃原子版税分润金库 |
| 会话授权层 | `LingheSessionKeyManager` | `0x8402000000000000000000000000000000000001` | 免弹窗有界会话密钥：强制执行单笔上限、总额度上限、到期时间与白名单接收方 |
| 合规储备层 | `LingheComplianceReserveVault` | `0x8800000000000000000000000000000000000001` | `Cap. 656` 储备二叉 Merkle 根锚定（强制 $R \geq L$）与跨链意图交割不可篡改审计账本 |
| 全链互联层 | `LingheOmnichainSettler` | `0x8900000000000000000000000000000000000001` | `ERC-7683` 跨链意图秒级垫付海关（`TIP-403` 双向筛查）+ `7.8000`（`7.7500–7.8500`）联系汇率池 |
| 零知识桥层 | `LingheZkLightClientBridge` | `0x8900000000000000000000000000000000000002` | `BN254 Groth16` 零知识轻客户端状态根双线性配对核验、防重放空值器与金库释放 |
| 硬件可信层 | `LingheHardwareAttestationVerifier` | `0x8900000000000000000000000000000000000003` | `EIP-7951` NIST `P-256` 芯片厂商根证书（NVIDIA H100 / AWS Nitro）与 `PCR0` 飞地度量注册表 |
| 主权治理层 | `LingheSovereignTimelockMultisig` | `0x8900000000000000000000000000000000000004` | 香港（`HK-HKMA`）· 新加坡（`SG-MAS`）· 东京（`JP-FSA`）· 苏黎世（`CH-FINMA`）`3-of-4` 主权门限多签 |
| 内核预编译 | `TIP-403 Policy Registry` | `0x403c000000000000000000000000000000000000` | Rust 执行内核原生转账合规白名单与制裁黑名单策略注册表（亚微秒级同步拦截） |
| 内核预编译 | `Enshrined StablecoinDEX` | `0xdec0000000000000000000000000000000000000` | Rust 内核级中央限价订单簿（CLOB）稳定币与原生代币流动性池（`LUSD/LHC`、`LHKD/LHC`） |
| 内核预编译 | `2D Nonce Manager` | `0x4e4f4e434500000000000000000000000000000000` | `0x76` 交易类型二维并行随机数（`nonceKey` × `nonce`）通道状态管理器 |
| 内核预编译 | `Account Keychain` | `0xaaaaaaaa0000000000000000000000000000000000` | 原生账户密钥链与 WebAuthn / Passkey 硬件公钥委托授权管理预编译 |
| 内核预编译 | `Simplex Validator Config` | `0xcccccccc0000000000000000000000000000000000` | Commonware Simplex BFT 验证人委员会公钥与跨法域共识权重配置预编译 |
| 内核预编译 | `TIP Fee Manager` | `0xfeec0000000000000000000000000000000000000` | 多稳定币（`LUSD` / `LHKD`）原生代付区块 Gas 费率折算与清算预编译 |

| 层级分类 | 规范模块 / 预编译名称 | 链上精确地址或程序标识（**SSOT**） | 核心协议职责与关键安全约束 |
|---|---|---|---|
| 通用预编译 | `Canonical Permit2` | `0x000000000022d473030f116ddee9f6b43ac78ba3` | 跨链意图与会话签名按单授权标准合约，消除无限额 `approve` 残留风险 |
| 密码学预编译 | `BN254 ecAdd / ecMul / ecPairing` | `0x0000000000000000000000000000000000000000000000000000000000000006` `0x0000000000000000000000000000000000000000000000000000000000000007` `0x0000000000000000000000000000000000000000000000000000000000000008` | `alt_bn128` 椭圆曲线 $\mathbb{G}_1$ 点加、标量乘与四配对最优 Ate 双线性映射核验（`EIP-196` / `EIP-197`） |
| 密码学预编译 | `Osaka EIP-7951 P256VERIFY` | `0x0000000000000000000000000000000000000000000000000000000000000100` | NIST `secp256r1` 硬件安全模块与机密计算飞地原生 `160` 字节输入帧验签预编译 |
| 后创世守卫层 | `LingheSovereignSecurityGuardV5` | 后创世确定性部署（`ChainID: 88888`，绑定 `0x8800...0001` 与 `0x8900...0004`） | 强制时间锁 eta = $t_{\text{queue}} + \Delta_{\min}$、链上 `LINGHE_CAP656_HQLA_V1` Merkle 核验、低阶 `P-256` 校验与白皮书 `SHA-256` 存证 |
| 后创世守护层 | `LingheGuardianModuleV51` | 后创世确定性部署（`ChainID: 88888`，`ERC-7201` 槽位 `0xdbe89959208436fd943d6683d5f7b77771cb0cd56885781f185f57de7fd7b700`） | 无管理后门 `ERC-7579` Validator/Hook：`k-of-n` Passkey 恢复（强制 $\geq 48$ h 时间锁与 $O(1)$ `cancelRecovery`）及每代币 `24h` 滚动限额 |
| 外部源链托管 | `LingheOriginEscrow7683` | 外部五大 EVM 源链确定性部署（`ChainID: 1, 10, 56, 8453, 42161`） | 双向 `ERC-7683` 源链托管：`openProtected` 防抢跑、`refundGraceBuffer` 防竞态、Groth16/多签双轨清偿与不可暂停到期退本 |
| 外部源链托管 | `linghe_solana_escrow_7683` | Solana BPF 程序 `Lh7683Escrow11111111111111111111111111111111`（`ChainID: 900`） | SPL 稳定币 PDA 托管、内省 `Ed25519SigVerify` 系统指令核验 `3-of-4` 守护者证明与不可暂停到期退本 |

附录 B · APPENDIX B

# 核心协议数据包与密码学域分离字节级编码规范

为确保跨语言审计工具、零知识电路证明器与硬件安全模块在实现时具备比特级一致性，本附录给出五大核心协议原像与预编译指令帧的精确字节偏移（Byte Offset）编码规范。

## B.1 `LINGHE_CAP656_HQLA_V1` 储备叶子紧凑原像布局（`abi.encodePacked` -> `keccak256`）

设账本原语符号 UTF-8 字节长度为 $L_{\text{sym}}$（`"LHKD"` 与 `"LUSD"` 均为 `4` 字节）、参考托管机构 SWIFT BIC 字节长度为 $L_{\text{bic}}$（标准 BIC11 为 `11` 字节）、参考隔离账户脱敏掩码长度为 $L_{\text{mask}}$，则总原像字节长度 $N = 57 + L_{\text{sym}} + L_{\text{bic}} + L_{\text{mask}}$：

| 字节偏移区间（**Byte Offset**） | 字节宽度 | **Solidity /** 线格式类型 | 字段名称与编码语义 |
|---|---|---|---|
| `[0 .. 20]` | `21 B` | `string` (UTF-8 ASCII) | 域分离常量 `"LINGHE_CAP656_HQLA_V1"`（`0x4c494e4748455f4341503635365f48514c415f5631`） |

| 字节偏移区间（**Byte Offset**） | 字节宽度 | **Solidity /** 线格式类型 | 字段名称与编码语义 |
|---|---|---|---|
| `[21 .. 20 + L_sym]` | `L_sym B` | `string` (UTF-8 ASCII) | `assetSymbol`（如 `"LHKD"` 或 `"LUSD"`，`4` 字节无填充） |
| `[21 + L_sym .. 20 + L_sym + L_bic]` | `L_bic B` | `string` (UTF-8 ASCII) | `custodianBic`（`ISO 9362` 参考银行标识码，如 `"HSBCHKHHHKH"`） |
| `[21 + L_sym + L_bic .. N − 37]` | `L_mask B` | `string` (UTF-8 ASCII) | `accountMask`（参考破产隔离账户脱敏掩码，如 `"HKMA-TRUST-SEG-8801"`） |
| `[N − 36 .. N − 5]` | `32 B` | `uint256` (Big-Endian) | `valuationAtomic`（以 $10^6$ 原子精度计量的单账户隔离储备法币估值） |
| `[N − 4 .. N − 1]` | `4 B` | `uint32` (Big-Endian) | `maturityDays`（储备资产剩余到期天数，即期存款为 `0`） |

### B.2 `LINGHE_HW_ATTESTATION_V3` 硬件可信飞地证明摘要原像（定长 `168 Bytes` -> `SHA-256`）

`LingheHardwareAttestationVerifier.computeHardwareQuoteDigest` 将如下 `168` 字节紧凑序列送入 `SHA-256` 预编译（`0x02`），生成 `32` 字节 `quoteDigest`：

| 字节偏移区间 | 字节宽度 | 字段类型与字节序 | 字段名称与链上绑定语义 |
|---|---|---|---|
| `[0 .. 23]` | `24 B` | `string` (ASCII) | 域分离前缀 `"LINGHE_HW_ATTESTATION_V3"` |
| `[24 .. 55]` | `32 B` | `uint256` (Big-Endian) | `block.chainid`（固定为 `88888` = `0x00...015b38`，阻断跨链重放） |
| `[56 .. 75]` | `20 B` | `address` | 验证合约地址（创世注册表为 `0x8900000000000000000000000000000000000003`） |
| `[76 .. 107]` | `32 B` | `bytes32` | `vendorId`（右填充硬件厂商标识，如 `"NVIDIA-H100-CC"`） |
| `[108 .. 139]` | `32 B` | `bytes32` | `pcr0Measurement`（治理白名单批准的可信飞地镜像密码学度量值） |
| `[140 .. 159]` | `20 B` | `address` | `runtimeSigner`（飞地内部生成并驻留于加密内存的运行时签名公钥地址） |
| `[160 .. 167]` | `8 B` | `uint64` (Big-Endian) | `validUntil`（硬件证明到期 UNIX 秒级时间戳，要求 `> block.timestamp`） |

### B.3 `LINGHE_PROTECTED_CLIENT_ID_V5` 防抢跑客户端订单标识原像（定长 `133 Bytes` -> `keccak256`）

`LingheOriginEscrow7683.computeProtectedClientOrderId(user, rawClientSalt)` 构造如下定长 `133` 字节原像并取 `keccak256`，使公开内存池观察者无法抢先注册受害者的业务盐值：

| 字节偏移区间 | 字节宽度 | 字段类型与字节序 | 字段名称与防抢跑绑定语义 |
|---|---|---|---|
| `[0 .. 28]` | `29 B` | `string` (ASCII) | 域分离前缀 `"LINGHE_PROTECTED_CLIENT_ID_V5"` |
| `[29 .. 60]` | `32 B` | `uint256` (Big-Endian) | `simulatedOriginChainId`（外部源链 Chain ID，如 `1`、`10`、`56`、`8453`、`42161`） |
| `[61 .. 80]` | `20 B` | `address` | 当前源链 `LingheOriginEscrow7683` 托管金库合约地址 |
| `[81 .. 100]` | `20 B` | `address` | `user`（订单真实发起人地址，确保不同调用者传入相同 `rawClientSalt` 派生出正交订单 ID） |
| `[101 .. 132]` | `32 B` | `bytes32` | `rawClientSalt`（客户端业务侧幂等随机盐值） |

### B.4 `0x0100` Osaka `EIP-7951` `P256VERIFY` 预编译调用帧与低阶标量约束（定长 `160 Bytes` 输入）

调用 `0x0000000000000000000000000000000000000100` 前，`LingheSovereignSecurityGuardV5.verifyCanonicalP256Signature` 先在合约层断言 $0 < r < n_{\mathrm{P256}}$ 且 $0 < s \le \lfloor n_{\mathrm{P256}}/2 \rfloor$，随后经 `staticcall` 提交定长 `160` 字节输入帧：

| 字节偏移区间 | 字节宽度 | 字段类型与字节序 | 字段名称与代数约束 |
|---|---|---|---|
| `[0 .. 31]` | `32 B` | `bytes32` (Big-Endian) | `messageSha256`（待验签消息的 `SHA-256` 摘要，如 `quoteDigest`） |
| `[32 .. 63]` | `32 B` | `uint256` (Big-Endian) | `r`（ECDSA 签名第一标量，满足 $1 \le r < 0$xFFFFFFFF00000000FFFFFFFFFFFFFFFFBCE6FAADA7179E84F3B9CAC2FC632551） |
| `[64 .. 95]` | `32 B` | `uint256` (Big-Endian) | `s`（规范低阶签名第二标量，满足 $1 \le s \le 0$x7FFFFFFF800000007FFFFFFFFFFFFFFFDE737D56D38BCF4279DCE5617E3192A8） |
| `[96 .. 127]` | `32 B` | `uint256` (Big-Endian) | `qx`（NIST `secp256r1` 公钥仿射横坐标 $Q_x \in \mathbb{F}_{p_{256}}$） |
| `[128 .. 159]` | `32 B` | `uint256` (Big-Endian) | `qy`（NIST `secp256r1` 公钥仿射纵坐标 $Q_y \in \mathbb{F}_{p_{256}}$；验签成功返回 `32` 字节 `0x00...01`） |

### B.5 Solana `Ed25519SigVerify` 指令内省头部与 `213 Bytes` 跨链清偿摘要布局

在 Solana 源链托管程序 `linghe_solana_escrow_7683` 中，`claim_by_solver` 通过系统指令内省账户（`Sysvar1nstructions1111111111111111111111111`）扫描同一交易中前置的原生验签程序 `Ed25519SigVerify111111111111111111111111111` 指令载荷：

| 结构分段与字节偏移 | 字节宽度 | 字节序与类型 | 字段定义与合约内省安全断言 |
|---|---|---|---|
| `Header [0]` | `1 B` | `u8` | `num_signatures`（本指令携带的签名总数 $K \ge 1$） |

| 结构分段与字节偏移 | 字节宽度 | 字节序与类型 | 字段定义与合约内省安全断言 |
|---|---|---|---|
| `Header [1]` | `1 B` | `u8` | `padding`（强制校验 `data[1] == 0x00`，阻断非规范编码畸变） |
| `Descriptor i [2+14i .. 3+14i]` | `2 B` | `u16` (Little-Endian) | `signature_offset`（强制 $\geq 2+14K$ 且 `+64 <= data.len()`） |
| `Descriptor i [4+14i .. 5+14i]` | `2 B` | `u16` (Little-Endian) | `signature_instruction_index`（强制 `== 0xFFFF`，仅允许本指令内联数据） |
| `Descriptor i [6+14i .. 7+14i]` | `2 B` | `u16` (Little-Endian) | `public_key_offset`（强制 $\geq 2+14K$，切片匹配 `guardian_ed25519_pubkeys[4]`） |
| `Descriptor i [8+14i .. 9+14i]` | `2 B` | `u16` (Little-Endian) | `public_key_instruction_index`（强制 `== 0xFFFF`） |
| `Descriptor i [10+14i .. 11+14i]` | `2 B` | `u16` (Little-Endian) | `message_data_offset`（强制 $\geq 2+14K$，切片必须等于 `expected_msg`） |
| `Descriptor i [12+14i .. 13+14i]` | `2 B` | `u16` (Little-Endian) | `message_data_size`（强制精确等于 `32`） |
| `Descriptor i [14+14i .. 15+14i]` | `2 B` | `u16` (Little-Endian) | `message_instruction_index`（强制 `== 0xFFFF`，阻断跨指令载荷借用） |

上述 `expected_msg` 为如下定长 `213` 字节跨链清偿原像的 `keccak256` 摘要：

| 原像偏移区间 | 字节宽度 | 字段类型与字节序 | 字段名称与跨链绑定值 |
|---|---|---|---|
| `[0 .. 32]` | `33 B` | `bytes` (ASCII) | `"LINGHE_SOLANA_ESCROW_7683_FILL_V4"` |
| `[33 .. 40]` | `8 B` | `u64` (Big-Endian) | `LINGHE_DESTINATION_CHAIN_ID`（固定为 `88888`） |
| `[41 .. 60]` | `20 B` | `[u8; 20]` | `LINGHE_SETTLER_EVM_ADDR`（`0x8900000000000000000000000000000000000001`） |
| `[61 .. 92]` | `32 B` | `[u8; 32]` | `order_id`（源链 PDA 托管金库绑定的唯一订单标识） |
| `[93 .. 124]` | `32 B` | `Pubkey` | `solver`（接收源链 SPL 稳定币清偿款的求解器公钥） |
| `[125 .. 144]` | `20 B` | `[u8; 20]` | `linghe_recipient`（目标链实际收款人 EVM 地址） |
| `[145 .. 164]` | `20 B` | `[u8; 20]` | `linghe_output_token`（目标链垫付资产 `LUSD` 或 `LHKD` 地址） |
| `[165 .. 172]` | `8 B` | `u64` (Big-Endian) | `output_amount`（目标链实际交割原子金额） |
| `[173 .. 180]` | `8 B` | `u64` (Big-Endian) | `linghe_block_height`（目标链包含交割交易的确定性区块高度） |
| `[181 .. 212]` | `32 B` | `[u8; 32]` | `linghe_tx_hash`（目标链交割交易哈希；位掩码去重满 `3-of-4` 方可放款） |

附录 C · APPENDIX C

# 国际标准、监管法例与学术参考文献


1. Chan, B. Y., & Pass, R. (2023). *Simplex Consensus: A Simple and Fast Consensus Protocol*. In Theory of Cryptography Conference (TCC 2023), Lecture Notes in Computer Science, vol. 14371, pp. 452–479. Springer.
2. Groth, J. (2016). *On the Size of Pairing-Based Non-interactive Arguments*. In Advances in Cryptology — EUROCRYPT 2016, Lecture Notes in Computer Science, vol. 9666, pp. 305–326. Springer.
3. Reitwiessner, C. (2017). *EIP-196: Precompiled contracts for addition and scalar multiplication on the elliptic curve alt_bn128*. Ethereum Improvement Proposals.
4. Buterin, V., & Reitwiessner, C. (2017). *EIP-197: Precompiled contracts for optimal ate pairing check on the elliptic curve alt_bn128*. Ethereum Improvement Proposals.
5. Johnson, R., Chow, W., & Bendiab, A. (2017). *EIP-712: Typed structured data hashing and signing*. Ethereum Improvement Proposals.
6. Buterin, V., Conner, E., Dudley, R., Slipper, M., Norden, I., & Bakhta, A. (2019). *EIP-1559: Fee market change for ETH 1.0 chain*. Ethereum Improvement Proposals.
7. Zoltu, M. (2020). *EIP-2718: Typed Transaction Envelope*. Ethereum Improvement Proposals.
8. Uniswap Labs & Across Protocol Working Group. (2024). *ERC-7683: Cross Chain Intents Standard (`IOriginSettler` & `IDestinationSettler`)*. Ethereum Request for Comments.
9. Buterin, V., Wilson, S., Dietrichs, A., & lightclient. (2024). *EIP-7702: Set EOA account code*. Ethereum Improvement Proposals.
10. Dubois, R., et al. (2025). *EIP-7951 (RIP-7212): Precompile for `secp256r1` Curve Support at `0x0100`*. Ethereum Osaka Execution Specification.
11. Trustless Agents Standards Working Group. (2025). *ERC-8004: Sovereign Agent Identity, Verifiable Reputation, and Inference Validation Registries*. Ethereum Request for Comments.
12. Fielding, R., Nottingham, M., & Reschke, J. (2022). *RFC 9110: HTTP Semantics (Section 15.5.3: 402 Payment Required)* & *Open HTTP `x402` Machine-to-Machine Micropayment Protocol Specification*. IETF Standards Track.
13. National Institute of Standards and Technology (NIST). (2023). *FIPS PUB 186-5: Digital Signature Standard (DSS) — Elliptic Curve Digital Signature Algorithm over Prime Curve P-256 (`secp256r1`)*. U.S. Department of Commerce.
14. Josefsson, S., & Liusvaara, I. (2017). *RFC 8032: Edwards-Curve Digital Signature Algorithm (EdDSA)*. Internet Engineering Task Force (IETF).
15. Hong Kong Special Administrative Region Government. (2025). *Stablecoins Ordinance (Chapter 656 of the Laws of Hong Kong / 香港法例第 656 章《稳定币条例》)* & HKMA Supervisory Guideline on Licensed Stablecoin Issuers.
16. Hong Kong Special Administrative Region Government. *Companies Ordinance (Cap. 622)*, *Anti-Money Laundering and Counter-Terrorist Financing Ordinance (Cap. 615, Section 13A)*, & *Personal Data (Privacy) Ordinance (Cap. 486)*.
17. Hong Kong Monetary Authority (HKMA). (2024). *Hong Kong's Linked Exchange Rate System (LERS): Convertibility Undertakings (`7.7500–7.8500 HKD/USD`) and Exchange Fund Bills & Notes (EFBN) Framework*. HKMA Monetary Management Publication.
18. European Parliament and Council of the European Union. (2023). *Regulation (EU) 2023/1114 on Markets in Crypto-Assets (MiCA), Titles III (Asset-Referenced Tokens) and IV (E-Money Tokens)*. Official Journal of the European Union.
19. Financial Action Task Force (FATF). (2023). *Updated Guidance for a Risk-Based Approach to Virtual Assets and Virtual Asset Service Providers — Recommendation 16 (Wire Transfers / Travel Rule)*. FATF Paris.
20. Basel Committee on Banking Supervision (BCBS). (2024). *Basel Framework — SCO60: Prudential Treatment of Cryptoasset Exposures & Disclosure of Cryptoasset Exposures*. Bank for International Settlements (BIS).

21. Bank for International Settlements (BIS) Innovation Hub, HKMA, Bank of Thailand, Digital Currency Institute of the PBC, & Central Bank of the UAE. (2024). *Project mBridge: Connecting Economies through CBDC and Multi-Jurisdictional PvP Settlement*. BIS Publications.
22. Monetary Authority of Singapore (MAS) & BIS Innovation Hub. (2024). *Project Guardian: Institutional Asset Tokenisation, Enshrined Compliance, and Interoperable Financial Networks*. MAS Financial Markets Development Series.




VOL. I · FORMAL PROTOCOL & CRYPTOGRAPHIC SPECIFICATION

250 ms 单槽确定性终局 · 二叉默克尔储备核验原语 · 协议内生合规状态机

链上密码学核验凭据与协议研发主体登记卡

FROZEN SSOT · v2.0

协议与软件研发主体

**MYRILUM LIMITED (HK Cap. 622 Body Corporate)**

公司注册与商业登记证号

CR/UBI: 80912065 · BR: 80912065-000-07-26-0

主权网络与链标识

ChainID: 88888 (0x15b38)

后创世链上存证守卫

LingheSovereignSecurityGuardV5.
latestWhitepaperAnchor()

不可变创世零块 **SHA-256**

96208e9900b218935143be3024d2602566a86d61caa906d42399f35b83afd991

本卷 **Markdown** 母本 **SHA-256**

5bf2bec72cdda421c44854027d6776ff9596d11d72762c93caa5024828a32818

确定性在线核验端点

/v1/whitepapers/verify

香港法定注册办事处

UNIT 1603, 16/F THE L. PLAZA, 367-375 QUEEN'S RD CENTRAL, SHEUNG WAN, HK

© 2026 MYRILUM LIMITED. All Rights Reserved. HK Cap. 656 & EU MiCA Zero-Yield Settlement Architecture · Zero Individual PII