

接口文档

1. 区块链底层组 (Blockchain Core)

成员 1: 区块链核心开发 (数据层)

核心职责: 定义数据结构, 维护链的完整性。

阶段	接口名	输入	输出	处理逻辑
数据定义	Transaction, Block, Account (类定义)	无	Transaction, Block, Account 类	定义 Pydantic 模型或 Python 类。
哈希计算	calculate_hash, get_merkle_root	任意字符串/字节流	哈希值 (String)	实现 SHA256 哈希算法; 实现 Merkle Tree 根哈希计算。
链维护	Blockchain.add_block	Block 对象 (来自成员 2 的挖矿结果)	更新后的区块链列表	验证区块哈希链接 (prev_hash); 将合法区块追加到 self.chain 列表。
公钥管理	PublicKeyRegistry.get_public_key	地址 (String)	公钥 (String)	根据账户地址查找对应的公钥, 供成员 2 进行签名验证使用。
地址生成	generate_address	公钥 (bytes)	地址 (String)	实现从公钥生成账户地址的算法 (如 SHA256 + 截取)。

成员 2: 区块链核心开发 (执行层)

核心职责: 交易执行与状态更新。

阶段	接口名	输入	输出	处理逻辑
交易池管理	Blockchain.add_transaction	Transaction 对象	待打包交易池	验证交易签名 (tx.verify()); 检查 Nonce 防止重放; 加入 pending_transactions。
挖矿/出块	Blockchain.mine_block	pending_transactions (内部状态)	新生成的 Block 对象	从池中取出交易; 调用 StateProcessor 执行交易; 打包生成新 Block。
状态更新	StateProcessor.apply_transaction	Transaction 对象	更新后的 WorldState	解析交易 Payload; 调用成员 3/4 的合约接口更新全局状态 (WorldState)。

2. 智能合约组 (Smart Contracts)

成员 3: 智能合约开发 (流程控制)

核心职责: 运维 SOP 状态机逻辑。

阶段	接口名	输入	输出	处理逻辑
SOP 状态流转	OpsContract.submit_analysis	Transaction (类型: SUBMIT_ANALYSIS , PROPOSE_ROOT_CAUSE)	更新后的 SOP 状态 (Data_Collected 等)	检查当前 SOP 阶段 (current_state) 是否允许该操作； 更新状态机。
事件记录	Event (数据结构)	业务操作结果	Event 对象 (供前端查询)	生成结构化日志， 记录关键步骤 (如 “分析已提交”)。

成员 4: 智能合约开发 (经济模型)

核心职责: Token 管理与共识计算。

阶段	接口名	输入	输出	处理逻辑
Token 管理	TokenContract.transfer , TokenContract.stake , TokenContract.slash	Transaction (类型: TRANSFER , STAKE)	更新后的 balances 字典	检查余额；执行转账/质押/ 扣除操作；更新账本。
共识投票	GovernanceContract.vote	Transaction (类型: VOTE)	提案状态 (Passed / Rejected)	记录投票；计算权重 (质押量 * 信誉分)； 判断是否达到阈值。

3. Agent 适配组 (Agent Adapter)

成员 5: Agent 中间件开发

核心职责: 链上交互中间件。

阶段	接口名	输入	输出	处理逻辑
交易封装	Transaction.sign (调用)	Agent 的决策 (Action , Payload)	签名后的 Transaction 对象	构造 Transaction 对象；调用 sign() 签名。
上链交互	ChainClient.send_action	Transaction 对象	交易回执 (Receipt)	调用成员 2 的 add_transaction 接口发送交易；轮询等待回执。

成员 6: Agent 适配与 Prompt 工程

核心职责: Agent 身份与 Prompt 改造。

阶段	接口名	输入	输出	处理逻辑
身份管理	AgentProfile (数据结构)	无	AgentProfile (包含地址、私钥)	生成/加载钱包私钥；为每个 Agent 实例分配身份。
Prompt 改造	System Prompt	原始运维任务描述	结构化的 Agent 输出 (包含 action_type , stake_amount)	修改 System Prompt，注入“质押”、“经济激励”等约束；解析 LLM 输出为 JSON。

4. 前端交互组 (Frontend & API)

成员 7: 前端开发 (区块链浏览器)

核心职责: 区块链数据可视化。

阶段	接口名	输入	输出	处理逻辑
后端 API 开发	GET /api/blocks , GET /api/transaction/{hash}	Blockchain 对象 (来自成员 1/2)	JSON 数据	封装 FastAPI 接口， 暴露区块和交易数据。
前端展示	BlockExplorer 组件	API 返回的 JSON 数据	渲染区块列表、 交易详情页、Merkle 树验证视图	调用 API 获取数据并渲染。

成员 8: 前端开发 (运维控制台)

核心职责: 业务状态与经济看板。

阶段	接口名	输入	输出	处理逻辑
后端 API 开发	GET /api/state/sop , GET /api/state/agents	OpsContract , TokenContract 状态 (来自成员 3/4)	JSON 数据	封装 FastAPI 接口，暴露 SOP 状态和 Agent 资产信息。
前端展示	SOPDashboard , EconomyMonitor 组件	API 返回的 JSON 数据	渲染 SOP 流程图 (React Flow)、 资产图表 (ECharts)	调用 API 获取数据并渲染。