《Subnet：暗流》升级版方案汇报

对齐 V4.3：实值 TAO 经济 · 在线竞技 · 数据飞轮｜含最小化 DEMO 交付路径

汇报目标	方案评审 + 对外同步（产品 / 业务视角）
范围	升级版技术方案（v3.1）＋最小化 DEMO （5 天 / 2 人）

1 | 项目定位：卖“信任生产流水线”

Undercurrent 不是卖算力 / 模型，而是通过经济博弈持续产出可信数据与 AI 审计能力。

一句话：运行在 Bittensor 上的功能型子网——将《暗流》博弈过程转化为可销售的监督数据与可复用审计机制。

1 监督数据包

JSON： 标签 + 元数据 + 证明 / 日志 + 哈希上链
面向： AI 公司 / 研究机构（订阅或按包购买）

2 审计规则模板

合约模板 + 部署脚本 +evidence pack 示例
面向：其他子网（一次性授权 + 维护）

3 机制配置包

游戏包 + 参数模板 + 复现脚本
面向： Web3 游戏 / 外部应用（授权费）

2 ｜机制亮点：验证成本漏斗 + 游戏 × 子网双向闭环

从“便宜→昂贵”渐进验证； AI 输出 risk_score 只驱动抽检，不参与最终裁判。

验证成本漏斗（ Stage 1 → 4 ）

Stage 1 静态检查

格式 / 签名 / 重复 /commit 校验（自动，极低成本）

Stage 2 基准评测

留出集 gating （中等成本）

Stage 3 风险驱动抽检

p_audit = p0 + k · risk_score （ AI 辅助，中等成本）

Stage 4 挑战仲裁

evidence pack 复现 + 深审（少量样本，昂贵）

双向经济闭环（核心飞轮）

AI 输出 risk_score （ 0-100 ）
用于驱动抽检概率

3 ｜系统全景架构：对外可消费，对内可验证

核心分层：客户端 → 数据存储 → 链下服务 → 子网运行时 / 合约 → API/ 市场网关 → 外部消费者。

外部消费者层
AI 公司 / Web3 游戏 / 其他子网 / NFT 市场

API & Market Gateway
鉴权｜计费｜路由｜跨子网结算

Bittensor 子网运行时 & 核心合约
质押｜数据提交｜验证评分｜奖励分配｜挑战仲裁｜声誉 NFT

链下服务层
AI 审计引擎｜ TEE 游戏服务｜ VDF Prover ｜数据索引器

数据存储层
Raw Data Pool(IPFS) ｜ Labeled Dataset(IPFS+SQL) ｜模型参数 Registry ｜审计日志

客户端层
《暗流》网页游戏｜外部游戏 Data SDK ｜本地 ZK Prover

对外输出形态

- 数据包（ JSON+ 证明）
- 审计 API （ REST/WS ）
- 规则模板 /SDK
- 声誉 NFT

4 ｜端到端工作流：一个 Epoch 内如何产出可信评分

从数据生成→提交→真值→验证→奖励→挑战→模型更新，形成可持续闭环。

注： AI 在早期“仅供参考”（陪审团引导期权重 0%），逐步过渡至混合验证。

关键机制对齐《暗流》 V4.3： 浅 / 深审计分层 + risk_score 风险引擎 + 标准化 evidence pack 导出

- 浅审计： 1 筹码 / 矿工，仅揭示信息（低成本侦查）
- 深审计： 3 筹码 / 矿工，揭示 + 罚款 + 广播（经济惩罚）
- evidence pack： action_log + labeled_data，支持挑战复现与训练闭环

5 ｜最小化 DEMO：两人 5 天交付“可玩的暗流”

目标：前后端并行，跑通一局（房间 / 角色 / 状态机 / 结算 / 导出 / 报告页）。

DEMO 目标 & 验收口径

- 2 人（前端 + 后端） 5 天内完成
- 7 人局可跑：不足人数由 Bot 补齐
- 阶段完整：声明→评分→审计→分配→交易→终局揭示→投票→结算
- 数据导出： raw-data.json / labeled-data.json + 报告页展示
- V4.3 对齐：浅 / 深审计、 AI risk_score （展示为概率条）、终局揭示惩罚

5 天并行计划（教程式 Step 0-10 ）

	后端	前端
Day 1	Step0-1 脚手架 + 共享类型	Step0-1 脚手架 + 共享类型
Day 2	Step2 后端规则引擎	Step3 前端路由 /Store/Socket
Day 3	Step4 REST+ 房间系统	Step6 分阶段界面组件
Day 4	Step5 首次联调握手	Step7 Bot+ 倒计时
Day 5	Step8-10 联调 + 导出 + 收尾	Step8-10 联调 + 报告页 + 动画

6 | DEMO 技术架构： Monorepo + Socket 状态推送

核心： shared/types 统一类型；服务端 GameEngine 驱动状态机；客户端以 PlayerViewState 渲染。

shared/types (前后端共用)

- Role / Phase / Session
- PlayerViewState 视图模型
- V4.3 数值常量 CONFIG

示例： export type GamePhase = 'lobby' | 'declaration' | ...

共享类型

server (Express + Socket.IO + SQLite)

- GameEngine： 状态机 + 筹码 / 审计 / 结算
- RoomManager： 房间 / 角色分配
- REST： rooms / export
- Socket： state:updated 推送

状态机（核心路径）
lobby → declaration → scoring → audit → distribution → trading → final_reveal → final_audit → final_vote → settlement

Socket/ REST

client (React + Vite + Zustand)

- Router： Home / Room / Game
- Store： 本地 UI 暂存 + 视图状态
- Socket Client： emit 操作 + receiv

state

-

按阶段组件： Declaration/Scoring/Aud t/...

渲染原则：服务端过滤视图
PlayerViewState 决定“每个角色看到什么”（矿工真值、验证者线索、所有者审计历史 …）

7 | DEMO 关键交互：可演示、可复盘、可导出

从“创建房间”到“终局报告”一条链路走通，并可用于后续子网数据格式对齐。

演示流程（建议讲解顺序）

1) 创建房间： POST /api/rooms → roomId
2) 加入房间： Socket room:join → 分配角色（Owner→Validator×3→Miner×4）
3) 开始游戏： game:start → 创建 Session → 推进到 declaration
4) 状态推送： state:updated （按角色过滤）
5) 各阶段操作： declare / score+report / audit(浅 / 深)+AI 风险条 / vote
6) 终局： final_reveal→final_vote→settlement （逐人揭示 +TAO 收益）
7) 导出： raw-data / labeled-data + ReportPage 展示映射

Socket 事件（核心最小集）

C→S	room:join / game:start
S→C	state:updated （单人）
C→S	player:declare / score / report
C→S	player:audit （浅 / 深） / ai_analysis
C→S	player:vote / kick / next_phase

对外同步话术： DEMO 不只是游戏原型，更是“数据格式 + 验证流程”的可视化沙盘。

8 ｜经济模型：子网运行时 × 游戏层（共享 TAO ）

游戏： Stake-to-Play ； 子网： API/ 数据经济；两层通过“收入注入 + 声誉 NFT” 闭环。

游戏层（ V4.3 对齐）

0.05 TAO 每人入场质押（ 7 人 =0.35 TAO ）

国库 5% 协议费后，剩余 95% 阶梯分账： 35% 25% 18% 4-5 名 15% 6 名 7% 7 名 0%

收益示例（池 =0.35 TAO ）

冠军	0.1164 TAO	+0.0664 （ +133% ）
亚军	0.0831 TAO	+0.0331 （ +66% ）
季军	0.0599 TAO	+0.0099 （ +20% ）

设计要点：局内“筹码”仅用于审计 / 举报 / 交易，不兑换 TAO （避免套利）。

子网运行时（链上）

- 质押门槛： Miner ≥15 TAO ｜ Validator ≥20 TAO
- Epoch 奖励默认： Validator 70% ｜ Miner 30% （可治理调整）
- 外部 API / 跨子网收入：建议 10% 注入游戏收益池（强化飞轮）
- 声誉 NFT ： 用于入场质押减免（最高 40% ）与跨子网可组合性

一句话：用外部收入“增厚奖金池”，用声誉 NFT “降低入场成本” → 形成稳定供给与增长。

9 | 风险与路线图：可控推进，先 DEMO 后子网

安全： ZK/VRF/VDF/ 挑战仲裁；路线： DEMO→ 测试网→混合验证→主网实值结算。

核心风险（ Top 4 ） & 防线

Miner 伪造 / 非真实对局数据

ZK 证明验证 GameStateTransitionCircuit ；无 ZK 自动扣分 / 惩罚

Validator 恶意评分 / 合谋

VRF 随机分配审查对象 + 偏差检测 + 举报 / 挑战机制

Ground Truth 被操纵

TEE 多节点一致性；不足时委员会 +VDF 时间锁 + commit-reveal

AI 模型投毒 / 偏差漂移

模型参数哈希上链 + 留出集盲测 + 人类陪审团监控 + 可回滚

路线图（阶段化）

Phase 0
DEMO 竞技原型（ M1-2 ）

Phase 1
测试网上线（ M3-4 ）

Phase 2
混合验证期（ M5-6 ）

Phase 3
完全自治 + 主网（ M7+ ）

下一步（建议本次评审确认）

1) DEMO 范围：是否按 5 天最小集交付（含导出 + 报告页）
2) 方案取舍： TEE 优先 or 先委员会路径（成本 / 速度）
3) 对外叙事：主推“可信数据产品”还是“审计 API”
4) 里程碑： Week10 DEMO / Week14 测试网 / Week28 主网（可按资源调整）