达芬奇公链：AI Agent 经济的结算层

Web4.0 时代的以太坊 Layer 2

2026 年 9 月
V0.1 工作稿

摘要：达芬奇公链（DaVinci Chain）是 Web4.0 世界中 AI Agent 经济的结算层，以一条专为 AI Agent 构建的以太坊 Layer 2 实现。它采用与 Robinhood Chain 相同的技术路线：基于 Arbitrum Orbit（Nitro）栈，以 Rollup 模式直接结算至以太坊，使用以太坊 blob 承载数据可用性，以 ETH 作为 gas 。在此之上，达芬奇叠加了一层通用 L2 所不具备的 Agent 原生协议层，包括链上身份、可编程账户与策略引擎、机器微支付、能力发现、执行验证与 Agent 间原子结算。人类用户在现有公链上是一等公民，Agent 不是。达芬奇的设计目标，是让 Agent 在链上拥有可验证的身份、可约束的资金权限与可结算的经济关系，即身份主权、资金主权与执行主权。六个子层收束于同一件事：让两个互不相识的 Agent 之间的第一笔交易能够安全结算 。本文分三部分：第一部分论证 Web4.0 的到来与当前 Agent 所缺失的基础设施，第二部分给出达芬奇的完整技术架构与设计取舍，第三部分说明网络经济学、治理与去中心化路径。文中凡涉及尚未上线的能力，均明确标记为路线图。

1. 从 Web1 到 Web4：第四次范式转移

互联网的每一次范式转移，都由同一个问题的答案改变所定义：谁是网络的主要行动者。

Web1 的行动者是发布者。少数机构生产内容，其余人只是接收者。网络在那时是一本可以翻阅但不能批注的书。

Web2 把这个角色交给了用户。社交网络、 UGC 平台与移动互联网让内容生产权下放到每一个人手里，代价是身份、数据与分发权集中到少数平台。用户创造了价值，却不拥有价值。

Web3 的行动者仍然是人，但人第一次在网络上拥有了可验证的资产与身份。公链把所有权从平台的数据库里取出来，写进一台任何人都可以独立验证的状态机。这是一次所有权的转移，但它没有动摇一个前提：链上的每一个动作，最终都要由一个人类按下确认。

Web4 改变的正是这个前提。当 AI Agent 能够理解目标、拆解任务、调用工具并连续执行多步计划，它们就不再只是人手里的工具。它们开始成为网络上独立的行动者，感知环境、进行推理、与其它 Agent 协作，并以自己的名义发生经济关系。

这不是一个关于遥远未来的判断。 2025 到 2026 年间，支撑 Agent 成为经济主体的几项关键标准集中落地：

- MCP（Model Context Protocol）统一了 Agent 与外部工具、 API 、合约之间的调用接口，回答了 Agent 如何使用世界；
- A2A 与 AP2 定义了 Agent 之间的通信与代理支付语义，回答了 Agent 如何与同类协作、如何代表用户付款；
- x402 把 HTTP 402 状态码复活为机器可解析的支付协议，让按次计费的微支付第一次具备工程可行性；
- ERC-8004（Trustless Agents）于 2025 年 8 月提出，2026 年 1 月在以太坊主网上线，为 Agent 提供了链上身份、信誉与验证的注册表标准。

工具、通信、支付、身份，四块地基在十八个月内先后落位。窗口已经打开。打开之后真正稀缺的，是一个把这些标准原生承载起来、并让它们彼此咬合的执行与结算环境。

表 1：互联网四代范式的行动者与所有权结构

范式	行动者	核心命题
Web1 · 读	发布者	信息可被获取
Web2 · 写	用户	人人可生产，平台占有分发
Web3 · 拥有	人	资产与身份归个体所有
Web4 · 自主	Agent	机器成为独立经济主体

要强调的是，Web4 并不取代 Web3，而是建立在它之上。 Agent 之所以能够成为可信的经济主体，恰恰因为公链提供了不依赖任何中介的所有权与结算基础。缺了这一层，Agent 的自主性就只能存在于某家公司的服务器里，那是托管，不是自主。

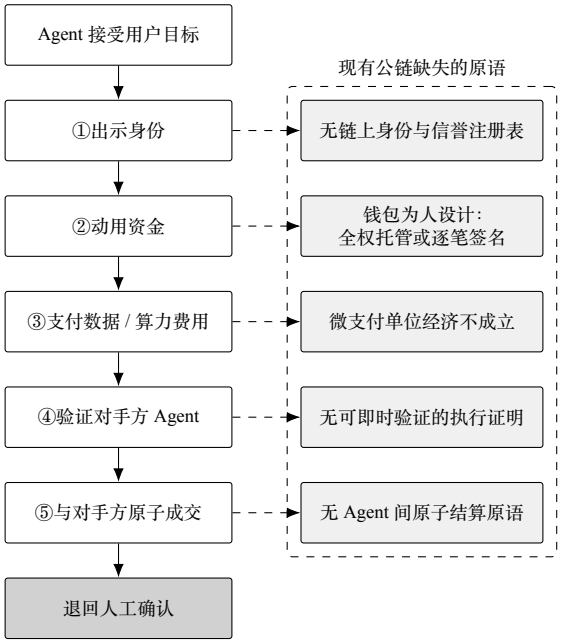
2. Agent 有了智能，但没有主权

今天的 AI Agent 已经能够理解目标、拆解任务、调用工具并连续执行多步计划。它缺的不是智能，是主权：

在一个开放网络里以自己的名义行动、持有价值并承担后果的资格。

把一次完整的 Agent 自主交易拆开看，这种资格的缺失会在五个位置上依次暴露出来。

图 1：一次 Agent 自主交易在现有公链上的五个断点



2.1 身份：*Agent* 无法证明自己是谁

人类用户的链上身份由私钥定义，这就够了，因为人可以在链下用社会关系、声誉与法律责任为自己背书。Agent 没有这些。一个地址背后是一个经过审计的执行体，还是一段随手部署的脚本，链上无从分辨。

后果是 Agent 之间无法建立任何先验信任。你不知道对面的 Agent 由谁部署、遵循什么策略、历史上是否履约。ERC-8004 提供了注册表标准，但标准需要一个把它当作系统底座而非可选合约的执行环境，否则身份只是一张没人查验的名片。

2.2 账户：钱包是为人设计的

现有账户模型的每一个假设都指向人类：一次交易对应一次签名，签名对应一次人的确认。这个假设对 Agent 完全不成立。一个持续运行的 Agent 每天可能发起成千上万次动作，逐笔弹窗确认在工程上等于让它停机。

于是实践中只剩两条路。要么把私钥直接交给 Agent，等于把全部资金置于一段代码的裁量之下；要么把资金留在中心化托管方，等于放弃 Web3 的全部意义。在全权与无权之间，缺少一个可编程的中间态。

2.3 支付：机器的账单结构与人不同

人类的支付是低频、大额、有心理阈值的。机器的支付是高频、微额、按调用计费的。一个 Agent 为一次模型推理支付 0.002 美元、为一条数据支付 0.0001 美元，一天发生数万次。这种账单结构在今天的公链上无法成立，因为单笔成本与最小可行金额都高出若干个数量级。

x402 给出了协议层的答案。但协议需要匹配的执行成本才能落地，微支付归根到底是个单位经济问题，不是协议问题。

2.4 信任：无法验证另一个 *Agent* 是否诚实执行

当 Agent 开始互相购买服务，买方面对的是一个黑箱：付了钱，对方是否真的按承诺跑了那次推理？用的是不是声称的模型？结果有没有被篡改？

人类市场用品牌、合同与法律解决这个问题，周期以月计。机器市场的交易在毫秒之间完成，等不了。它需要的是可即时验证的执行证明，以及有经济代价的信誉，而这两样在现有公链上都没有原生位置。

2.5 结算：*Agent* 与 *Agent* 之间没有成交轨道

最后一步最关键。两个 Agent 谈成一笔交换，一方提供数据、另一方支付费用，或者两个 Agent 互相对冲头寸，这笔交换需要原子性：要么同时发生，要么都不发生。

现有公链当然能做原子交换，但原语是为人类资产交易设计的，粒度粗、授权重，无法表达「在预算与约束条件内自主达成」这类机器意图。结果是绝大多数 Agent 之间的经济关系最终仍要退回到人类中介或中心化平台来结算，而那正是 Agent 经济试图绕开的东西。

这五个缺口不是五个独立的功能需求，它们是同一件事的五个侧面。现有公链把人当作一等公民，把 Agent 当作人的延伸。只要这个前提不变，Agent 就只能是一个更聪明的按钮。达芬奇要改变的正是这个前提。

3. 为什么现有公链装不下 Agent 经济

一个自然的反驳是：以太坊与它的 Layer 2 已经是图灵完备的通用计算平台，Agent 需要的一切都可以用合约实现，何必再造一条链?

这个反驳低估了默认值的力量。一条链能做什么，取决于它的虚拟机；一条链上实际会发生什么，取决于它的默认值，包括账户模型、费用结构、出块节奏、内置注册表与标准的可得性。现有公链的默认值是围绕人类用户校准的，而 Agent 与人在六个维度上存在结构性错配。

3.1 六个结构性错配

①账户模型面向人。一次交易对应一次签名的假设，把人的确认硬编码进了交易生命周期。账户抽象（ERC-4337）与 EOA 委托（EIP-7702）提供了出路，但它们在多数链上是应用层的可选项，不是系统底座。每个团队各自实现一套会话密钥与权限逻辑，互不兼容，也无法被第三方验证。

②费用结构不适配微支付。人类交易的经济学允许每笔几美分到几美元的成本。 Agent 按调用计费的经济学要求单笔成本低于所购买资源的价值，即亚美分乃至更低（达芬奇上这一量级的适用范围见 §8.3）。更麻烦的是波动：Agent 的预算是程序设定的，一次 gas 尖峰就足以让整条策略链路失效。

③无原生身份与信誉层。地址是匿名的字节串，没有任何与之绑定的能力声明、部署者信息或履约历史。ERC-8004 补上了标准，但标准若不是链的一部分，就只是一份没有强制力的约定。

④无意图与授权边界原语。用户对 Agent 的授权本质上是一句带约束的委托：在这个预算内、这些市场上、这个时间窗里，替我达成这个目标。链上没有表达这种约束的原语，于是约束只能写在链下的业务逻辑里，既不可验证，也不可强制。

⑤出块与确定性不匹配机器节奏。人类可以等待十几秒的确认，Agent 的决策循环以毫秒计。更要紧的是确定性：Agent 需要知道自己的动作何时不可逆，才能安全地推进下一步，而不是靠轮询和重试来猜测状态。

⑥现有「 AI 概念链」多为叙事包装。市场上已有不少自称面向 AI 的链，但绝大多数停留在把模型推理结果上链存证，或者发行一个与 AI 相关的代币，缺少 Agent 真正需要的身份、授权与结算能力。把 AI 写进名字，和把 Agent 写进协议，是两件事。

3.2 横向对比

表 2：主流执行环境对 Agent 原生能力的支持情况

执行环境	身份	策略账户	微支付	A2A 结算
以太坊 L1	标准可用	应用层	不可行	无
通用 L2	标准可用	应用层	勉强	无
高性能 L1	无	应用层	可行	无
「 AI 概念链」	多为存证	应用层	视实现	无
达芬奇	协议内置	协议内置	协议内置	协议内置

这张表要说明的不是别人做不到。通用 L2 当然可以部署 ERC-8004 注册表。它要说明的是位置差异：同样的能力，作为可选合约存在，和作为协议底座存在，会导向完全不同的生态。前者让每个开发者重新发明一次轮子，后者让所有 Agent 天然共享同一套身份、授权与结算语义。

Agent 经济需要的不是更强的通用链，而是一条把 Agent 原语设为默认值的链。这是达芬奇的判断。

4. 达芬奇的答案：Agent 主权栈

达芬奇把第 2 章的五个缺口归纳为三层主权，并以此组织整条链的设计。

4.1 三层主权

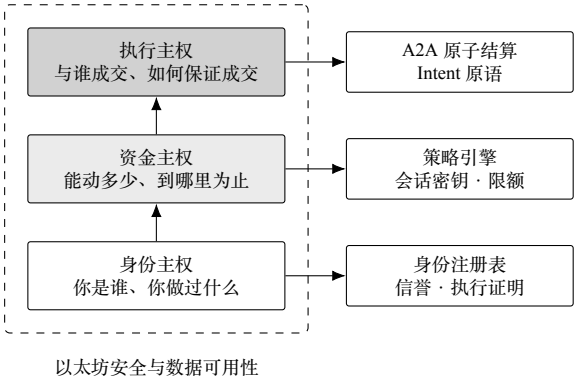
身份主权（Identity Sovereignty）。 Agent 拥有独立于部署者的链上身份。这个身份承载它的能力声明、部署来源与履约历史，可被任何第三方在不接触部署者的前提下验证。它解决的是「你是谁、你做过什么」。

资金主权（Capital Sovereignty）。 Agent 可以在用户授予的明确边界内独立支配资金。边界由链上策略强制执行，不是由链下代码自觉遵守；用户可随时以一笔链上交易收回授权。它解决的是「你能动多少、动到什么程度为止」。

执行主权（Execution Sovereignty）。 Agent 可以与其它 Agent 直接达成并结算经济关系，不需要人类在中间按下确认，也不需要中心化平台做对手方。它解决的是「你能和谁成交、怎样保证成交」。

三者是递进的。没有身份就无法建立信任，没有资金边界就不敢授予自主，没有结算轨道则自主无处施展。三层齐备，Agent 才从人的延伸变成独立的经济主体。

图 2：Agent 主权栈与对应的协议能力



4.2 三条设计原则

Agent-first：**把 Agent 当作默认用户**。凡在「对人更友好」与「对机器更友好」之间存在取舍的设计，达芬奇选择后者。人类用户通过 Agent 与网络交互，而不是反过来让 Agent 模拟人类的交互方式。

Ethereum-aligned：**安全性不自建**。达芬奇不发明新的共识，不自建验证者网络，不要求生态为一条新链的安全性重新承担风险。它是以太坊的 Layer 2，安全性与数据可用性来自以太坊，流动性与开发者工具来自 EVM 生态。创新应当发生在协议层，不是共识层。

Standards-native：**采纳标准，而非创造方言**。 ERC-8004 、 ERC-4337 、 EIP-7702 、 x402 、 MCP 、 A2A 已经是事实标准。达芬奇的工作不是提出替代方案，是把它们从可选合约提升为协议底座，并为它们提供匹配的执行成本与结算能力。一个在达芬奇上写的 Agent，应当能与以太坊生态里的任何 Agent 对话。

4.3 关于名字

列奥纳多 · 达 · 芬奇留下的大部分机械设计，飞行器、自动纺车、可编程小车，在他生活的年代都无法制造。缺的不是构想，是承载构想的材料与工艺。

今天的 AI Agent 处在相似的位置。智能已经具备，缺的是让它落地的基础设施。

5. 架构总览

达芬奇是一条分层系统。每一层只解决一类问题，把自己不擅长的问题交给下一层：安全性向下继承，性能在中间获得，差异化在上层构建。

图 3：达芬奇公链的六层架构

L1 · 以太坊安全层。达芬奇不自建共识，不自建验证者集合。状态的最终性、争议的裁决与资产的根本归属全部锚定在以太坊上。这意味着攻击达芬奇的成本等价于攻击以太坊的成本，而不是攻击一个新网络的代币市值。

L2 · 数据可用性层。每一批交易数据以 blob 形式发布到以太坊。任何人都可以据此独立重建达芬奇的完整状态，不需要征得排序器同意，也不需要信任任何委员会。

L3 · 结算与桥接层。资产的跨层进出、状态根的提交、欺诈证明的挑战窗口与逃生舱，都在这一层的以太坊合约中实现。用户资金的所有权路径完全独立于达芬奇的运营方。

L4 · 执行层（Nitro）。 EVM 等价的执行环境，负责交易排序、执行与本地状态维护。这一层决定用户感知到的速度与成本。

L5 · Agent 原生协议层。达芬奇与通用 L2 的全部差异集中在这一层：身份、账户与策略、支付、发现、验证、结算六个子层，以协议底座而非可选合约的形式提供（第 8 章）。

L6 · 应用生态层。由第三方开发者构建的 Agent 、Agent 市场与终端应用。基金会提供基础设施与公共品，不自营应用（第 11 章）。

分层的原则可以概括为一句话：凡涉及资金所有权的，锚定在以太坊；凡涉及机器交互体验的，做进协议底座；凡涉及具体业务形态的，留给生态。

5.1 为什么把自己称作结算层

六个子层听起来是六件并列的事，实际上它们有主次。

结算是唯一直接产生经济后果的动作。身份回答跟谁结算，账户回答能结算多少，支付提供结算的轨道，发现让双方找到彼此，验证决定这笔结算该不该发生。前五层都不创造价值，它们存在的意义是让第六层可以被信任地执行。

所以达芬奇把自己定位为 **AI Agent 经济的结算层**，而不是六个功能的集合。判断这条链是否成功的标准只有一条：两个互不相识的 Agent 之间的第一笔交易，能不能在没有人类中介、没有中心化平台做对手方的前提下安全完成。

需要区分的是，这里说的结算不限于支付。 Agent 之间的经济关系包括数据换支付、服务换费用、头寸互相对冲、意图的原子履行——支付只是其中最简单的一种。一条只做支付的链解决不了「对方是否真的按承诺执行」这个问题，而那恰恰是机器市场最贵的摩擦。

6. 底层选型：Arbitrum Orbit 以太坊 L2

6.1 为什么是 *Layer 2*，而不是自研 *L1*

一条新公链最昂贵的东西不是代码，是安全性。自研 L1 意味着要从零建立一个验证者集合、为它设计激励，并说服市场相信这套激励在长期博弈下不会崩溃。这个过程通常需要数年与巨额补贴，而在它完成之前，链上任何有价值的资产都处在不相称的风险之中。

Agent 经济对安全性的要求只会更高。当资金由程序而非人支配、交易在毫秒之间连续发生，人工干预的窗口实际上消失了。把安全性建立在以太坊上，是达芬奇能为 Agent 提供的第一个保证。

作为 L2，达芬奇同时继承三样东西：以太坊的安全预算、 EVM 生态的开发者与工具链、以及与整个以太坊流动性的直接连通。这些都不是一条新 L1 可以在短期内复制的。

6.2 为什么是 *Arbitrum Orbit*

在以太坊 L2 的技术栈中，达芬奇选择 Arbitrum Orbit（Nitro）栈，理由有三。

这条路线已被机构级实践验证。 Robinhood 于 2026 年 7 月上线的 Robinhood Chain 正是基于 Arbitrum Orbit 的以太坊 L2 。一家受严格监管、承担真实客户资产的上市金融机构选择这套技术栈并投入生产，是对其工程成熟度与运营可行性的有力背书。在基础设施选型上，被验证过通常比新颖更值钱。

Nitro 提供 EVM 等价，而不只是 EVM 兼容。达芬奇上的合约、工具、审计方法与以太坊主网一致，开发者不需要学习新的方言，安全审计的经验可以直接迁移。对一条希望承载资金的链而言，这一点的价值高于任何性能指标。

链参数高度可配置。 Orbit 允许在数据可用性模式、出块时间、排序机制、预编译与费用结构上做定制，这正是达芬奇把 Agent 原语做进协议底座所必需的自由度（第 7 、 8 章）。

6.3 Nitro 的关键技术特性

欺诈证明与挑战期。状态根提交到以太坊后进入挑战窗口，任何诚实参与者都可以对错误状态发起挑战，并通过交互式细分证明其错误。安全性依赖「至少一个诚实验证者」，不依赖对运营方的信任。

Stylus：EVM 之外的执行选项。 Nitro 支持以 Rust 、 C 、 C++ 编译到 WASM 的合约，与 Solidity 合约共享同一状态与地址空间，在计算密集型任务上有显著的成本优势。这对 Agent 侧的验证、密码学运算与策略计算是一条现实的优化路径。

可压缩至约 100 毫秒的出块时间。机器的决策循环以毫秒计，出块节奏直接决定 Agent 能否在链上完成紧耦合的多步操作。

6.4 一个必须澄清的事实

达芬奇是以太坊的 Layer 2，不是 Arbitrum 的 Layer 3 。它使用 Arbitrum 的技术栈，但父链是以太坊：状态根直接提交到以太坊，数据直接发布到以太坊的 blob 空间，安全性直接来自以太坊。这与 Robinhood Chain 的定位一致。

这个区别不是措辞问题。 L3 的安全性经由 L2 中转，其最终性、退出路径与故障域都多一层依赖；L2 则直接锚定在以太坊上。对一条要承载自主资金的链来说，少一层依赖就是少一类风险。

7. 链参数与设计取舍

本章逐项说明达芬奇的关键参数选择。每一项都给出选择、理由与代价，因为没有免费的设计。

7.1 数据可用性：*Rollup* 模式 + 以太坊 *blob*

选择。全部交易数据以 blob 形式发布到以太坊，采用完整的 Rollup 模式，不使用 AnyTrust 或外部 DA 委员会。

理由。数据可用性决定了一件事：当排序器失效或作恶时，用户能否在无需任何人许可的情况下重建状态并取回资产。Rollup 模式下这个能力是无条件的，DA 委员会模式下它依赖于委员会的诚实与在线。Agent 持有的资金由程序支配、缺少人工兜底，这类保证不应打折。

代价。blob 的数据成本高于外部 DA 方案。达芬奇接受这一成本，并通过批量压缩与执行层优化来摊薄它。如果未来出现需要极致低成本的特定场景，可以在不改变主网保证的前提下，以侧通道形式单独评估。

7.2 Gas 代币：*ETH*

选择。网络的 gas 以 ETH 计价与支付，与 Robinhood Chain 一致。DVC 不承担 gas 职能。

理由。其一是兼容性。ETH 作为 gas 意味着任何以太坊钱包、桥、工具与 Agent 框架都可以零改造接入，不需要为「先获取一种新代币」设计引导流程。其二是稳定性。Agent 的预算是程序设定的，一个价格波动剧烈的 gas 代币会让成本模型失效。其三是中立性：把网络的运行成本与代币的市场表现解耦，网络不会因代币价格波动而变得难用或不可预测。

代价。达芬奇主动放弃了 gas 需求这条最常见的代币价值捕获路径。DVC 的价值必须完全来自协议层的真实职能，即质押与罚没、服务结算、排序权与治理（第 12 章）。这是一个刻意的取舍：宁可让代币的价值论证更难写，也不愿让网络的使用体验更难用。

补充：Agent 不需要持有 ETH。通过协议层的稳定币 paymaster，Agent 可以用稳定币支付 gas，由 paymaster 代付 ETH。对 Agent 而言，账单是以稳定币计价的可预测成本；对网络而言，gas 仍然是 ETH。两个目标并不冲突。

7.3 出块时间与确定性

选择。目标出块时间为约 100 毫秒量级，并向 Agent 明确暴露三种状态：已接收、已排序（软确认）、已在以太坊上最终确认。

理由。Agent 需要的不只是快，更是知道什么时候可以安全地推进下一步。把确定性等级显式暴露为可编程的状态，比笼统地宣称一个延迟数字更有工程价值。

代价。软确认在以太坊最终确认之前仍受重组窗口约束。达芬奇不承诺软确认等同于最终性，并要求高价值结算路径显式等待最终确认。

7.4 排序器

选择。主网启动阶段由单一排序器运行，同时在第一天就提供强制包含（force inclusion）与逃生舱机制。排序器的去中心化按第 13 章的阶段路线推进。

理由。诚实地说，单排序器是当前所有主流 L2 的起点，它在启动阶段提供了必要的可运维性。关键不在于是否存在单点，而在于这个单点能否作恶，以及用户能否绕过它。

代价。在完成去中心化之前，排序器具有交易排序权与短期审查能力。达芬奇通过强制包含机制限定其上限：任何被排序器忽略的交易，用户都可以通过以太坊合约强制上链。

7.5 跨链桥：采用官方规范桥

选择。资产的跨层进出使用 Arbitrum Orbit 栈自带的官方规范桥（canonical bridge）。达芬奇不自建桥，也不把任何第三方桥作为协议推荐的主路径。

理由。桥是 L2 生态历史上损失最惨重的攻击面，过去数年因跨链桥被攻破而流失的资产以十亿美元计，绝大多数事故的根因相同：桥的安全性由一组独立于两条链之外的验证者或多签持有，一旦这组外部信任集合被攻破，两端的链本身再安全也无济于事。

官方规范桥不引入这样的外部信任集合。存款由 L1 的网关合约锁定并在达芬奇上铸造，提款提交后进入挑战期，期满由用户在 L1 直接领取。整条路径的安全性来自 L1 合约与欺诈证明，与链本身同源。用户不需要为跨链这一个动作额外信任任何人。

这一点对 Agent 尤其重要。Agent 的资金进出由程序发起，没有人工复核的环节，因此进出路径的行为必须完全由合约定义、可被静态验证，而不是取决于某个外部验证者集合当下是否诚实与在线。规范桥、强制包含与逃生舱走的是同一套 L1 合约路径（I-2 、 I-3），这让资金的进、出与紧急退出共享同一个安全假设。

代价。提款需要等待挑战期，这是乐观 Rollup 的固有属性，时长通常以天计。生态可以提供第三方流动性桥作为快速通道，达芬奇不阻止，但也不背书：使用这类通道意味着接受其自身的信任假设，风险由用户自行承担。

7.6 参数一览

表 3：达芬奇主网关键参数

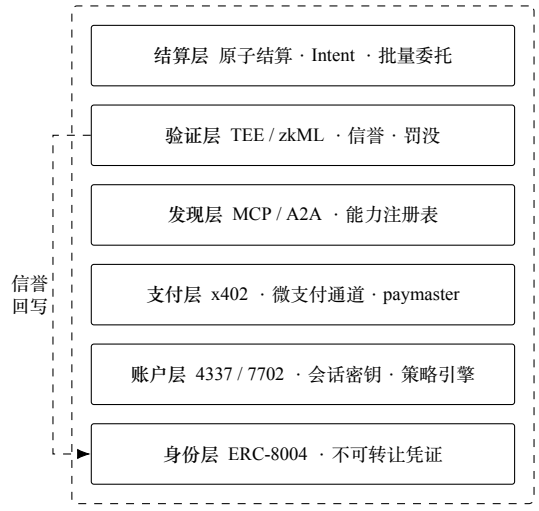
参数	取值
父链	以太坊（非 Arbitrum L3）
技术栈	Arbitrum Orbit / Nitro
证明模式	欺诈证明（交互式细分）
数据可用性	以太坊 blob（Rollup 模式）
Gas 代币	ETH（Agent 侧支持稳定币 paymaster）
出块时间	约 100 毫秒量级
执行环境	EVM 等价 + Stylus（WASM）
排序器	单一起步，按阶段去中心化
跨链桥	Arbitrum 官方规范桥（canonical bridge）
退出保证	强制包含 + 逃生舱，永久有效

8. Agent 原生协议层

这一章描述达芬奇与任何通用 L2 的全部差异。

差异不在于达芬奇能做到什么别人做不到。 EVM 是图灵完备的，任何合约都可以部署在任何 EVM 链上。差异在于位置：达芬奇把 Agent 所需的六类原语放进协议底座，让它们成为全网共享的默认语义，而不是每个应用各自实现、互不兼容的可选合约。

图 4：Agent 原生协议层的六个子层



以协议底座提供，而非可选合约

8.1 身份层：*Agent* 是谁

标准基础：ERC-8004 。达芬奇把 Trustless Agents 注册表实现为协议级注册表，由系统合约维护。任何 Agent 在链上活动前都可以注册一个身份，生态也默认预期它这么做。

身份包含什么。一个 Agent 身份绑定四类信息：能力声明（提供什么服务、接受什么输入）、部署来源（谁部署、基于什么代码指纹）、履约历史（由验证层写入，见 §8.5），以及经济担保（质押了多少 DVC，见 §12）。

不可转让性。 Agent 身份以不可转让凭证的形式存在，不能被买卖。这一点很重要：如果信誉可以被交易，那么信誉就不再是信誉，而是一种可以购买的伪装。

为什么必须在协议层。身份的价值与采用率成正比。一个只有某个应用内部承认的身份，对跨应用的 Agent 协作毫无用处。只有当身份是链的一部分、被所有应用默认查验时，它才能成为协作的基础。

8.2 账户层：*Agent* 能动多少

标准基础：ERC-4337 与 EIP-7702 。达芬奇原生支持账户抽象与 EOA 委托，Agent 的每一次动作都通过智能账户发起，不通过裸私钥。

会话密钥。用户向 Agent 授予的不是私钥，是一把有边界的会话密钥。这把钥匙在时间窗口内可以独立签名，无需用户逐笔确认，这正是让 Agent 得以连续运行的前提。

链上策略引擎。这是达芬奇账户层的核心。授权的边界不写在 Agent 的代码里靠它自觉遵守，而是写在链上由协议强制执行：

- 额度约束：单笔上限、累计上限、时间窗内的速率上限；
- 范围约束：可交互的合约白名单、可调用的方法、可持有的资产；
- 时间约束：授权的起止时间，过期自动失效；
- 复合约束：跨多次委托的风险累积上限，防止 Agent 通过递归授权绕过单笔限制；
- 即时撤销：用户可以用一笔链上交易立即收回全部授权，不需要 Agent 配合。

为什么必须在协议层。如果策略引擎是应用层合约，它的正确性需要被逐个审计，它的语义在不同应用间不一致，用户面对的是十几套互不相同的授权界面。把它做进协议，用户对任何 Agent 的授权都使用同一套可验证的语义。这就是全权与无权之间那个缺失的中间态。

8.3 支付层：*Agent* 如何付费

标准基础：x402 。达芬奇原生支持 x402 的支付语义，让「请求资源、收到报价、支付、获得资源」成为一

次可编程的机器交互，不需要预先建立账户关系。

微支付通道。对高频小额的重复调用，达芬奇提供状态通道原语：双方一次开启、链下累计、择机结算。这把单次调用的边际成本压到接近于零，使按次计费的机器账单在经济上成立。

成本分层，以及"亚美分"这个说法的适用范围。需要把话说准，因为这一条最容易被一支计算器证伪。达芬奇上存在三种成本量级，它们相差两个数量级：

表 4：三种支付路径的单次成本量级

路径	gas	单次成本
通道内调用（链下累计）	摊薄	$\to 0$
链上简单转账 / 直接支付	21,000	$\approx$ \$0.0011
链上复杂交互（DeFi 类）	250,000	$\approx$ \$0.013

按 L2 gas price 约 0.013 gwei 、 ETH 约 \$4,000 估算。这两个参数都会变，成本随之线性变动，所以上表给的是量级而非承诺值。

结论直接说：**亚美分成立于前两种路径，不成立于第三种**。一笔 250,000 gas 的链上复杂交互约 \$0.013，高于一美分，任何声称它是亚美分的说法都是错的。

这正是通道存在的理由，而不是通道的附加价值。一条通道把一次结算的链上成本摊薄到 N 次调用上，边际成本趋近于结算成本除以 N；当一个 Agent 一天调用某个数据源数万次时，N 足够大，单次成本在任何合理的 gas 假设下都可以忽略。反过来说，如果一类交互本质上必须逐笔上链且逻辑复杂，它就不属于微支付场景，达芬奇也不会假装它是。

稳定币 paymaster 。网络的 gas 以 ETH 计价（§7.2），但 Agent 可以用稳定币支付，由协议层的 paymaster 完成兑付。 Agent 的成本模型因此是可预测的稳定币金额，不受 ETH 价格波动影响。

按次计费原语。数据、推理、算力、 API 调用的计价与结算被抽象为统一的原语。服务提供方只需声明价格，调用方只需发起调用，计费与结算由协议完成。

8.4 发现层：*Agent* 如何找到彼此

标准基础：MCP 与 A2A 。 MCP 定义 Agent 如何调用工具，A2A 定义 Agent 之间如何通信。达芬奇提供链上的能力注册表：服务提供方登记自己的能力描述、价格、服务等级与端点，调用方按需检索。

链上索引，链下传输。需要诚实说明的是，Agent 之间的实际通信发生在链下，把每一次消息交换上链既不必要也不经济。链上承载的是可发现性与可问责性：谁提供什么、承诺了什么，以及承诺是否兑现（由 §8.5 记录）。

8.5 验证层：如何确认 *Agent* 诚实执行

这是 Agent 经济里最难、也最关键的一层。

执行证明。达芬奇支持两类证明路径。基于可信执行环境（TEE）的证明适用于对延迟敏感、对硬件信任可接受的场景；基于零知识的机器学习证明（zkML）适用于对信任要求更高、可接受更高证明成本的场景。二者并非互斥，是按场景选择的两档保证。

信誉注册表。每一次服务交付的结果，是否按时、是否符合声明、是否被争议，都写入与身份绑定的信誉记录。信誉是累积的、不可转让的，并与经济担保挂钩。

质押与罚没。信誉若没有经济后果，就只是一串数字。提供服务的 Agent 需要质押 DVC 作为履约担保，被证实的违约行为触发罚没，罚没的一部分补偿受损方。信任在这里从一种感觉变成了一个价格。

诚实的边界。达芬奇不声称能验证一个 Agent 的决策是否正确，那是一个无法定义的问题。它验证的是声明与执行的一致性：你声称用了某个模型、某份数据、某套参数，链上可以证明你确实用了。

8.6 结算层：*Agent* 之间如何成交

Agent 间原子结算。两个 Agent 达成的交换，无论是数据换支付、服务换费用还是头寸互相对冲，都在一个区块内原子完成：要么全部成立，要么全部回滚，不存在一方交付而另一方违约的中间状态。

Intent 原语。达芬奇把意图做成一等公民。 Agent 提交的不是一串具体操作，是一个带约束的目标：在什么预算内、满足什么条件、达成什么结果。协议负责校验最终执行是否落在约束之内，不落在约束内的执行被拒绝。

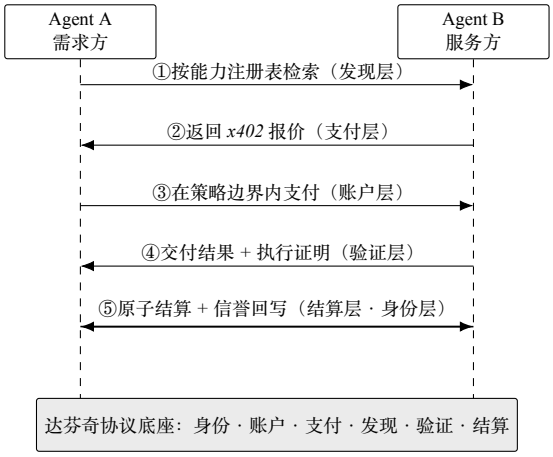
批量委托。一组时序相关的操作可以作为一个原子单元提交，避免部分成交造成的中间风险敞口。

结构化拒绝码。每一笔被拒绝的操作都附带机器可解析的原因码，Agent 据此自动调整而不是盲目重试。这是一个小设计，但它决定了 Agent 能否在无人值守的情况下稳定运行。

8.7 六层如何咬合

单独看，每一层都只是对一个已有标准的实现。它们的价值来自咬合。

图 5：一次完整的 Agent 间自主交易



一个 Agent 凭身份被发现（发现层），在策略边界内动用资金（账户层），以微支付购买服务（支付层），要求对方提交执行证明（验证层），并与对方原子结算（结算层）。交付结果反过来写入对方的信誉（身份层）。

这个闭环就是 Agent 经济的最小可运行单元。它在今天的任何一条通用公链上都无法在没有中心化中介的前提下完整跑通，这正是达芬奇存在的理由。

9. 智能供给原语：数据、模型与算力的链上市场

9.1 Agent 竞争的胜负手

当网络上运行着成千上万个 Agent，它们之间的差距来自哪里？

不来自算力。算力可以按量租用，价格公开、供给充足。也不来自基础模型，顶级模型的能力差距正在快速收敛，且大多可通过 API 获得。

差距来自输入：喂给 Agent 的数据是否及时、是否独家、是否干净，调用的模型是否针对任务微调，执行的策略是否经过验证。同样的模型，喂进垃圾数据只能输出垃圾决策。在 Agent 经济里，供给侧的质量就是竞争力本身。

问题在于这些高质量的输入今天几乎全部锁在封闭系统里。拥有独家数据的机构没有安全的方式把它卖给陌生的 Agent：买方不肯先付款，卖方不肯先交付，双方都无法验证对方是否诚实。这是一个典型的、可以被链解决的问题。

9.2 达芬奇提供原语，而非市场

一个定位说明：达芬奇在协议层只提供原语，不自营市场，与第 11 章的生态口径一致。

协议层提供五件事：

- 登记。供给方在能力注册表中声明自己提供什么，数据源、模型、算力或策略，以及服务等级；
- 计价。以统一的按次计费原语表达价格，支持一次性购买、订阅与用量阶梯；
- 结算。买卖双方的交付与付款在一个区块内原子完成，消除先付款还是先交付的信任死结；
- 存证。供给物的指纹上链，让「你卖给我的是不是你声称的那份」成为一个可验证的问题；
- 问责。交付质量写入供给方的信誉，违约触发质押罚没（§8.5）。

在这五件事之上，具体的数据市场、模型市场、算力市场由生态建设。达芬奇的角色是让这些市场可以被建立，而不是自己去当其中一个。

9.3 防抄袭与可验证交付

数据与模型是可复制品，这让它们的链上交易面临一个特有的困难：买方一旦拿到，就可以无成本地转售。

达芬奇在原语层提供两个工具应对。

指纹存证。供给物的密码学指纹在交付前上链。任何后续出现的副本都可以被追溯到原始供给方与首个购买方，转售无法匿名。

凭证化访问。对持续供给，比如实时数据流，交付的不是数据本身而是一把有边界的访问凭证：绑定身份、限定时间与用量、可被撤销。这把「一次性卖出一份拷贝」变成「持续提供一种服务」，从根本上改变了可复制品的商业结构。

需要诚实说明的是，这两个工具降低抄袭的收益与匿名性，但不能在密码学上杜绝抄袭。任何声称能彻底解决可复制品盗版问题的设计都值得怀疑。达芬奇的目标是让诚实交易比抄袭更划算，不是让抄袭在物理上不可能。

10. 安全模型与不变量

把资金交给一个由程序自主支配的 Agent，是一件需要理由的事。本章给出达芬奇的理由：一组不变量，即协议设计的硬约束。任何违反它们的代码变更都应被

治理拒绝，任何运行时检测到的违反都应触发相应的保护动作。

10.1 资金主权不变量

I-1 ｜用户资金路径独立于运营方。排序器的运营方对用户资金没有托管权。所有资产锁定在以太坊上的合约中，运营方可以被更换、排序器可以被选举化，而资金的所有权路径不因此改变。

I-2 ｜逃生舱永久有效。若排序器停止服务超过约定窗口，任何用户都可以直接通过以太坊上的合约退出，取回其资产。这一能力不依赖运营方配合，也不可被治理关闭。

I-3 ｜强制包含不可剥夺。任何被排序器忽略或审查的交易，用户都可以通过以太坊合约强制上链。排序器拥有排序权，但没有否决权。

10.2 Agent 行为不变量

I-4 ｜ Agent 不可越过授权边界。由 Agent 发起的每一笔交易必须落在用户声明的策略约束之内（§8.2）。越界的交易由协议拒绝，不依赖 Agent 自觉遵守。

I-5 ｜ Agent 不可通过递归委托累积无界风险。协议追踪 Agent 跨多次委托的风险复合，触及阈值自动降级或冻结。这封堵了「每笔都合规、总量却失控」这一类最隐蔽的失效模式。

I-6 ｜ Agent 的每一笔交易可复现、可审计。 Agent 的签名、触发条件与所依赖的链上状态快照均上链，任何第三方都可以在任意区块高度重放其执行路径并独立验证。

I-7 ｜授权可被即时单方面撤销。用户收回授权不需要 Agent 配合，也不需要等待任何窗口期。一笔链上交易立即生效。

10.3 协议运营不变量

I-8 ｜关键参数变更须经治理时间锁。费用结构、策略引擎语义、验证规则、罚没条件等核心参数的任何修改，必须经链上治理且设有时间锁。在锁定期内，不认同变更的用户可以自由退出。

I-9 ｜升级权限逐步交出。主网启动阶段存在的紧急权限，其范围、持有者与移除时间表必须公开，并按第 13 章的阶段路线逐步销毁。临时的中心化需要一个明确的失效日期，否则它就不是临时的。

10.4 风险披露

诚实地列出达芬奇当前与可预见的风险。

排序器中心化风险。在完成去中心化之前，单一排序器具有排序权与短期审查能力。缓解措施是强制包含与逃生舱（I-2 、 I-3），但用户体验在排序器故障期间会显著劣化。

欺诈证明的活性依赖。乐观 Rollup 的安全性依赖「至少一个诚实验证者在挑战期内发起挑战」。达芬奇将资助独立验证者并开源验证工具，但这一假设本身无法被消除。

验证层的信任边界。 TEE 证明依赖硬件厂商的信任根，存在侧信道与固件漏洞的历史先例；zkML 证明成本高、适用模型范围有限。达芬奇提供两档保证，但不声称任何一档是无条件的。

Agent 自身的失效。协议能保证 Agent 不越过授权边界，但不能保证 Agent 在边界内做出好的决策。一个在授权范围内持续亏损的 Agent，是用户的风险，不是协议的缺陷。这个边界必须被清楚地传达给用户。

标准演进风险。 ERC-8004 、 x402 等标准仍在演进，未来版本可能与当前实现不兼容。达芬奇将以可升级的适配层承接标准变更，但迁移成本无法完全避免。

监管风险。自主 Agent 代表用户持有并支配资产，在多数司法辖区尚无明确的法律定性。达芬奇作为中立的协议层不对参与方做任何身份审查（§11），相应的合规责任由参与方自行承担。

11. 生态建设与开发者计划

11.1 基金会的角色边界

先说清楚一件事：达芬奇基金会建设基础设施与公共品，不自营应用。

基金会的资金不会投入到与生态开发者竞争的产品上。这不是一种姿态，是一条自我约束。一条链如果同时是它上面最大的应用，开发者就必须时刻担心自己在为未来的竞争对手做验证。

基金会做三件事：把协议做好，把工具做全，把早期开发者扶起来。应用形态由生态决定。

11.2 场景蓝图

以下场景说明协议层如何被使用，不是基金会的产品计划。

Agent 自主金融。 Agent 代表用户在明确的风险预算内执行交易、做市、资产管理与套利。协议层的支撑是：策略引擎约束风险敞口（§8.2），原子结算消除对手方风险（§8.6），执行证明使业绩归因可验证（§8.5）。这是对策略边界与结算原子性要求最高的场景，也因此最能检验协议设计。

Agent 商务与自动采购。 Agent 代表个人或企业发现供给、比价、议价并完成采购。协议层的支撑是：能力注册表提供可检索的供给侧（§8.4），x402 让按次付费无需预建账户关系（§8.3），信誉与罚没为陌生对手方提供先验信任（§8.5）。

机器对机器的资源结算。算力、数据、 API 、推理服务在 Agent 之间按调用计费、按秒结算。协议层的支撑是：微支付通道把单次调用的边际成本压到接近于零（§8.3），凭证化访问把可复制品变成可持续计费的服务（§9.3）。

Agent-as-a-Service 。 Agent 开发者把自己的 Agent 作为可被调用的服务发布，按使用量而非按软件授权收费。协议层的支撑是：不可转让的身份与信誉让优秀 Agent 的声誉可累积（§8.1），质押与罚没过滤低质量供给（§8.5）。

11.3 开发者基础设施

基金会承诺提供并持续维护：

- Agent SDK：身份注册、策略账户创建、支付与结算的完整封装，覆盖主流语言；
- MCP 接入套件：让既有的 MCP Agent 能以最小改动接入达芬奇的身份、支付与结算能力；
- Agent 模板库：策略账户、微支付通道、执行证明提交等常见模式的参考实现，经审计后开源；
- 测试网与水龙头：与主网参数一致的公共测试网；
- 审计补贴：为生态项目的安全审计提供资助，降低早期团队的安全门槛；
- 标准共建：作为 ERC-8004 、 x402 等标准的实现方与贡献方参与上游演进，而非另立方言。

11.4 无许可与中立

达芬奇是一条无许可的公链。协议层不对任何参与方做身份审查或地域限制，不设白名单，不保留封禁权限。用户、 Agent 、开发者、服务提供方都一样。

这是中立性的必然要求。一条可以决定谁能使用的链，不是基础设施，是一个平台。应用层可以按其自身需要实现任何合规模块，但那是应用的选择，不是协议的强制。相应的合规责任由各参与方自行承担。

11.5 生态不变量

前面几节讲的是基金会打算做什么。这一节讲的是基金会不能做什么，以及为什么这部分需要被写死。

政治经济学里有一个被反复验证的观察：一个稳定的社会会随时间持续累积分利集团，它们不创造增量，只争夺存量分配，最终把社会拖进一个低活力的均衡。奥尔森在《国家兴衰探源》里把这称为制度僵化，机制很朴素——这类集团的形成成本高，解散成本更高，所以数量只会单调累积。公链生态没有豁免权。生态基金运行几年之后，受益人名单高度重合、排放锁死在早已不产生增量的老池子里、失败的项目以战略重要性为由拿到追加拨款，这些都是同一种病的症状。

代币治理让情况比现实世界更棘手。现实中的利益集团要通过游说间接影响立法，而在代币投票下，拿到大额分配的主体同时就是决定下一轮分配的投票人。这不是游说，是自己给自己拨款。

针对性的制度设计已经有成熟答案。温加斯特等人在市场保护型联邦主义中指出，要让一个分权结构真正约束住各级的攫取行为，关键不在于分权本身，而在于两件事：统一市场内不允许存在排他性特权，以及所有参与方都面对硬预算约束。达芬奇把这两条连同退出权，写成六条生态不变量。

E-1 ｜基金会不自营应用。基金会资金不得投入与生态开发者竞争的产品。支出公开可审计。

E-2 ｜协议层无排他性特权。任何协议层能力，包括接口、费率、路由顺序与注册表位次，对所有参与方以同一条件开放。不存在为单个项目保留的专属通道。成熟的既得利益者索取的往往不是拨款而是排他权，因为拨款是一次性的，排他权是永久现金流。

E-3 ｜硬预算约束。 Grant 一次性拨付并与里程碑挂钩。同一主体不得获得救助性追加拨款。只要失败之后还能回来要钱这条路存在，理性策略就会变成先拿钱再游说续命，而不是把事做成。

E-4 ｜激励强制日落。任何激励与排放计划必须在设立时写明终止区块。续期需要重新走完整审批流程，不存在默认展期。这一条直接针对奥尔森机制里唯一的变量，也就是时间。

E-5 ｜利益冲突回避。拨款受益人及其已知关联地址不得对涉及自身的拨款提案投票。

E-6 ｜退出无成本。不设排他性集成、不可迁移积分或强制锁仓。生态项目与用户可以随时离开，而不损失已经获得的链上资产与身份。达芬奇采纳上游标准而不创造方言（§4.3），本身就是这条不变量在技术层面的实现：一个在达芬奇上写的 Agent 能迁到别处去。

表 5：生态不变量的强制层级

编号	内容	强制层级
E-1	基金会不自营应用	章程 · 支出可审计
E-2	无排他性协议特权	合约强制
E-3	硬预算约束，不救助	合约强制
E-4	激励强制日落	合约强制
E-5	利益冲突回避	合约部分强制
E-6	退出无成本	合约强制

关于这六条，有两点需要讲清楚，否则它们就只是好听的话。

第一，强制层级不能含糊。E-2 、 E-3 、 E-4 、 E-6 写进合约，违反即无法执行；E-1 是章程约束，靠公开支出审计问责；E-5 只能按已知关联地址集合排除，对付不了刻意构造的女巫。把政策承诺包装成硬约束，是白皮书最常见的失信来源，我们宁可把边界标清楚。

第二，约束按阶段生效。奥尔森描述的是成熟社会的病，不是新生社会的病。一条刚上线的链需要倾斜资源扶持早期建设者，第一天就上满全部约束只会让没有人来建。因此 E-3 与 E-4 在生态进入第 13 章的阶段二之后完全生效，在此之前以较宽的阈值运行。与紧急升级权限一样，临时的例外需要一个写在章程里的失效条件，否则它就不是临时的。

生态项目还有一个结构性弱点值得单独指出：它们通常没有自己的收入，几乎全部依赖基金会拨款，而这本身就是软预算约束的定义。合约条款解决不了这个问题。基金会的做法是在资助决策中优先支持能够自己产生收入的项目，并把拨款占其收入的比例作为续期评估的硬指标。这条比任何治理条款都更能决定生态的长期健康。

12. 网络经济学与 DVC 代币角色

本章说明网络如何在经济上自我维持，以及 DVC 在协议中承担什么职能。

前提必须先讲清楚：网络的 gas 以 ETH 计价与支付（§7.2），DVC 不承担 gas 职能。这意味着 DVC 的价值论证不能依靠「用链就要买币」这条最容易写、也最经不起推敲的路径，而必须落在协议层的真实职能上。

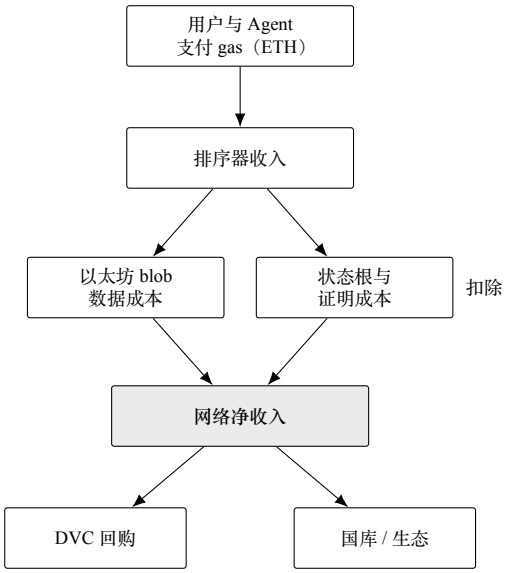
表 6：DVC 代币基本参数

项	取值
名称	DaVinci Coin
符号	DVC
精度	18
总供应量	1,000,000,000（10 亿）

增发与销毁机制、分配结构与释放安排以独立披露文件为准，不在本技术白皮书的范围之内。

12.1 网络费用流

图 6：达芬奇的网络费用流



用户与 Agent 支付的 gas 以 ETH 计入排序器收入。排序器需要为此支付两项成本：向以太坊提交 blob 数据的费用，以及状态根提交与证明相关的费用。二者相减即为网络净收入。

这个结构有两个值得注意的性质。其一，网络净收入与使用量直接相关，与代币价格无关，网络的可持续性不依赖于市场情绪。其二，Agent 的高频微交易特征意味着单笔利润极薄而笔数极大，这要求执行层的效率被持续优化，而不是靠提高费率来改善经济性。

12.2 DVC 的协议层职能

履约担保：质押与罚没。这是 DVC 最核心的职能。在链上提供服务的 Agent，包括数据供给方、模型提供方、执行服务方，必须质押 DVC 作为履约担保。质押规模决定其可承接的业务上限，被证实的违约触发罚没，部分补偿受损方。

这条职能的分量在于它是第 8.5 节整个信任层的经济地基。没有可罚没的抵押，信誉就只是一串无法执行的数字。有了它，陌生 Agent 之间的先验信任才有了价格。 Agent 经济的规模越大、对信任的要求越高，被锁定为担保的 DVC 就越多。

服务市场的结算媒介。能力注册表中登记的服务，其订阅、分成与担保以 DVC 计价结算（§9.2）。这里需要说明，达芬奇不强制所有交易必须使用 DVC，强制计价会损害可用性；协议为以 DVC 结算的路径提供更低的费率与更高的担保效率。

排序与验证的质押。随着排序器去中心化推进（第 13 章），参与排序与验证需要质押 DVC 并接受罚没约束。这把网络的活性与安全性同代币的经济价值绑定。

治理权。协议参数、验证规则、罚没条件、国库支出的决策权（第 13 章）。治理是伴随属性，不作为主要的价值论证。

净收入回购。网络净收入的一部分用于在公开市场回购 DVC，具体比例由治理决定并公开执行。这让网络的使用规模与代币形成结构性联系。

把上述五条放在一起，DVC 的需求来源可以概括为一句话：网络上需要被担保的经济活动越多，被锁定的 DVC 越多。这条论证依赖的是网络上真实发生的服务交易与担保需求，不是交易笔数乘以费率。

DVC 的总量、分配结构、释放曲线与解锁节奏以独立披露文件为准，不在本技术白皮书的范围之内。

13. 治理与去中心化路线

一条由单一实体运营的链，无论技术上多么精巧，都不具备承载自主经济的资格，因为 Agent 无法与一个可以随时改变规则的对手方建立长期关系。去中心化对达芬奇不是理想主义的装饰，是功能性的必需。

本章给出路径，而不是承诺一个日期。

13.1 排序器去中心化的三个阶段

表 7 给出各阶段对应的可验证里程碑。达芬奇按阶段而非日期交付，每个阶段以里程碑标识完成，不以时间承诺标识开始。

阶段一：单一排序器 + 无条件退出。主网启动形态。排序器由基金会运营，但强制包含与逃生舱在第一天即生效（I-2 、 I-3）。此阶段的安全模型是：排序器可以让你不方便，但不能让你损失。

阶段二：许可多方排序 + 强制轮换。引入多个独立运营方共同排序，出块权按规则轮换，任一方的审查行为可被其它方绕过。此阶段消除单点故障与单点审查，但参与者仍需准入。

阶段三：无许可排序 + 质押罚没。任何满足质押要求的参与方均可加入排序，恶意行为通过罚没惩罚。此阶段完成后，网络的活性不再依赖任何特定实体。

同步推进的是证明去中心化：验证者集合从基金会资助的少数节点，扩展到任何人都可以运行并因发现错误而获得奖励的无许可集合。

13.2 升级权限的处置

主网启动阶段存在紧急升级权限。这是所有负责任的新链的现实选择，因为早期漏洞的修复速度可能决定用户资产的存亡。达芬奇对这项权限做三点承诺：

- 公开：权限的范围、持有者构成与触发条件全部公开；
- 受限：权限只能用于安全事件响应，不能用于变更经济参数或没收资产；
- 有期限：按上述阶段路线逐步收窄直至销毁，每一次收窄公开可验证（I-9）。

13.3 治理的范围与边界

链上治理决定协议参数、验证规则与罚没条件、国库支出，以及排序与验证的准入规则。所有核心参数变更须经时间锁，锁定期内用户可自由退出（I-8）。

同样重要的是治理不能决定的事。治理不能没收用户资产，不能关闭逃生舱，不能取消强制包含，不能追溯修改已完成的结算。这些约束写在合约里，不是写在承诺里。

14. 结语

达芬奇公链要解决的问题可以用一句话概括：AI Agent 已经具备智能，但还没有在开放网络中以自己的名义行动、持有价值并承担后果的资格。

我们的答案不是发明一套新的共识或一种新的虚拟机。安全性向以太坊借，执行环境用已被机构级实践验证过的 Arbitrum Orbit 栈，协议标准采纳生态已有的 ERC-8004 、 ERC-4337 、 x402 、 MCP 与 A2A 。达芬奇真正做的事只有一件：把这些散落的标准，从各自为战的可选合约，变成彼此咬合的协议底座。

这件事听起来不够激动人心。但基础设施的价值从来不在于它有多新颖，而在于它有多可靠、多中立、多不容易被替换。 Web4 的机器经济需要的不是又一条讲故事的链，是一条让 Agent 之间的第一笔陌生交易能够安全完成的链。

达 · 芬奇在五百年前画下了他无法制造的机器。今天我们不缺构想，缺的是承载构想的工艺。

表 7：分阶段交付与里程碑

阶段	可验证里程碑
Phase 1 · Testnet	公共测试网上线；身份层与账户层（含策略引擎）可用；Agent SDK 首个版本开源
Phase 2 · Mainnet Genesis	主网上线；官方规范桥、强制包含与逃生舱生效；稳定币 paymaster 可用；核心合约审计报告公开
Phase 3 · Agent Protocol	支付层、发现层、验证层、结算层完整开放；智能供给原语上线；首批第三方 Agent 在主网运行
Phase 4 · Ecosystem	开发者计划规模化；能力注册表中的第三方服务达到规模；多方排序进入阶段二
Phase 5 · Sovereign	无许可排序与验证；紧急权限销毁；治理移交 DAO

披露

本文档为达芬奇公链技术白皮书 V0.1 工作稿。文中涉及的技术方案、参数与阶段规划可能随工程进展调整，最终以主网实际上线状态与官方公告为准。本文档中描述的能力，凡未在已完成阶段中列出的，均应视为路线图承诺，而非当前状态。

本文档不构成任何司法辖区下的证券发行要约或投资建议。 DVC 代币不被设计为任何司法辖区的证券。DVC（DaVinci Coin，精度 18，总供应量 1,000,000,000）的分配与释放安排不在本文档范围内，以独立披露文件为准。

达芬奇是无许可的中立协议层，不对任何参与方进行身份审查或地域限制，参与者需自行评估并承担所在司法辖区下的合规责任。参与公链生态涉及智能合约风险、加密资产波动风险与技术演进风险，参与前请阅读第 10.4 节的风险披露。

参考标准与实现

1. ERC-8004: Trustless Agents，Agent 链上身份、信誉与验证注册表标准。
2. ERC-4337: Account Abstraction Using Alt Mempool，账户抽象标准。
3. EIP-7702: Set EOA Account Code，外部账户委托标准。
4. EIP-4844: Shard Blob Transactions，以太坊 blob 数据可用性。
5. x402，基于 HTTP 402 的机器微支付协议。
6. Model Context Protocol (MCP)，Agent 与工具的标准化调用接口。
7. Agent-to-Agent (A2A)，Agent 间通信协议。
8. Arbitrum Nitro / Orbit，本链所采用的 Rollup 技术栈。
9. Robinhood Chain，采用同一技术栈的以太坊 L2，2026 年 7 月主网上线。
10. Mancur Olson, *The Rise and Decline of Nations*, 1982 。分利集团与制度僵化，第 11.5 节的诊断依据。
11. Montinola, Qian & Weingast, “Federalism, Chinese Style”, *World Politics*, 1995 。市场保护型联邦主义，第 11.5 节的制度设计依据。