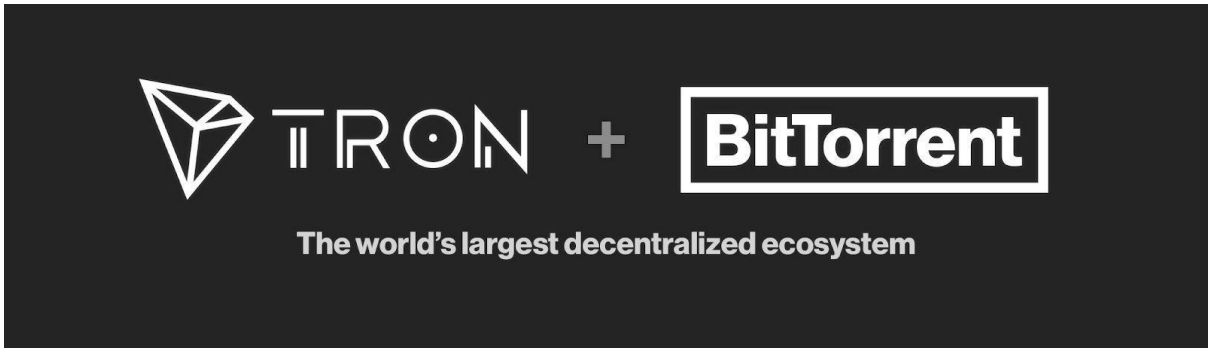


BitTorrent (BTT) 白皮书

Oct. 2026

1. 引言 **4**

2. 项目背景与动因 **4**

2.1 大文件、模型与数据分发的需求 4

2.2 AI 算力的供需问题 5

2.3 链上结算需求 5

2.4 BTT 生态的设计原则 5

3. 生态架构 **6**

3.1 分发层：BitTorrent 6

3.2 去中心化算力层：BTTInferGrid 6

3.3 网络与结算层：BTTC 6

4. 产品一：BitTorrent 生态的“流量与分发引擎” **7**

4.1 设计原则 7

4.2 P2P 分发协议与网络原语 8

4.2.1 内容寻址与完整性验证 8

4.2.2 对等发现与组网 8

4.2.3 分片交换与带宽分配 9

4.2.4 传输与拥塞控制 10

4.3 客户端节点网络 10

4.4 BitTorrent Speed：基于 BTT 的带宽激励层 11

4.4.1 问题定义：做种激励的结构性缺失 11

4.4.2 总体设计 12

4.4.3 竞价机制 13

4.4.4 撮合与拍卖 13

4.4.5 支付通道与结算 14

4.4.6 从带宽到通用资源交换 15

4.5 面向模型与数据集的分发能力 16

4.5.1 负载特征与适配性 16

4.5.2 工件分发约定 17

4.5.3 分发模式 18

4.5.4 长尾可用性与激励 18

4.5.5 定位边界 19

5. 产品二：BTTInferGrid 生态的“去中心化算力网络” **19**

5.1 产品定位：生态的计算与存储层 19

5.2 参与角色与资源接入机制 20

5.3 AI 推理任务与算力调度流程 20

5.4 可验证推理与服务质量机制 21

5.5 分布式存储与数据服务能力 21

5.6 BTT 结算、节点激励与惩罚机制 21

5.7 开发者接口与应用接入 22

6. 产品三：BTTC 链上执行与生态结算网络 22

6.1 产品定位、用户与场景 22
6.2 架构：三大功能层、关键参与角色 23
6.3 机制：服务执行与链上结算流程 24
6.4 机制：共识机制与安全模型 24

7. BTT 经济模型与代币设计 25

7.1 核心代币：BTT 25
7.2 BTT 代币分布 25
7.3 质押与奖励机制 27
7.3.1 验证人质押机制 27
7.3.2 验证人合伙机制 28
7.3.3 委托质押机制 28
7.4 解锁与释放时间 29
7.5 通胀模型 30
7.6 回购与销毁机制 30

8. 应用场景与生态构建 31

8.1 去中心化 AI 推理与存储服务 31
8.2 模型、数据集与内容的 P2P 分发 31
8.3 跨链桥与资产流通 32
8.4 DAO 治理与投票系统 32

9. 路线图与生态发展规划 32

9.1 阶段路线图 32
9.2 生态发展重点 33
9.3 安全与治理 33

10. 结语 33

1. 引言

随着数字内容规模持续增长、生成式 AI 快速发展以及 Web3 应用不断扩张，现有基础设施仍面临内容与模型分发效率不均、算力资源供需分散，以及链上服务成本较高、规模化结算能力不足等问题。

BitTorrent 生态定位为涵盖内容分发、去中心化算力以及链上扩容与结算的基础设施体系，由三个核心产品和统一的代币经济机制组成：

- **BitTorrent**：负责点对点内容传输，连接用户、节点与带宽资源；
- **BTTInferGrid**：面向 AI 推理服务，负责算力资源的接入、调度、执行与验证；
- **BTTC**：定位为面向异构公链的 Layer 2 扩容网络与基础设施，提供高性能、低成本的智能合约执行和链上结算环境；
- **BTT**：作为贯穿生态的经济媒介，用于资源支付、网络 Gas、节点激励、质押与治理。

BitTorrent 生态通过整合分布式内容传输、去中心化算力、链上执行能力及统一经济机制，致力于构建服务 Web3 与 AI 应用的开放、高效、可持续的去中心化基础设施网络。

2. 项目背景与动因

2.1 大文件、模型与数据分发的需求

AI 模型和大型数据集通常达到数十 GB 甚至数百 GB。依赖中心化服务器分发，需要承担较高的带宽和流量出口成本，在集中下载时还可能出现源站拥堵、访问中断及区域可用性不足等问题。文件传播过程中也面临版本混乱、内容被篡改和来源难以确认的风险。

BitTorrent 依托长期积累的点对点内容传输能力，连接全球用户、节点与带宽资源。在此基础上，BitTorrent 将进一步面向 AI 模型、数据集及其他大型数字内容，提供开放、高效的分布式分发服务。

2.2 AI 算力的供需问题

生成式 AI 的快速发展带动了推理算力需求持续增长，但大量 GPU 和计算资源仍分散在不同地区、设备及服务平台中。算力供需双方缺乏高效的连接方式，部分计算资源处于闲置状态，而开发者仍面临算力获取成本较高、资源调度不够灵活和服务质量难以验证等问题。

BTTInferGrid 面向 AI 推理服务，致力于连接全球分散的 GPU 算力资源，建立开放、可验证的去中心化算力网络。通过资源接入、任务调度、推理执行和结果验证等机制，为开发者及 AI 应用提供更加灵活的算力支持。

2.3 链上结算需求

随着 Web3 应用和去中心化资源服务不断发展，链上交互、资产流转和服务结算需求也在持续增加。不同公链之间相互隔离，主链在成本和处理能力方面存在差异，使大规模、跨网络的应用协作面临一定限制。

BTTC 定位为连接异构公链的 Layer 2 扩容网络与基础设施，为智能合约执行、跨链交互和链上结算提供高性能、低成本的网络环境，并为 BitTorrent 生态内的数据、算力和存储服务提供价值流转支持。

2.4 BTT 生态的设计原则

BitTorrent 生态围绕内容分发、AI 算力与存储以及链上基础设施构建多层次产品体系，整体设计遵循以下原则：

- 分层协同： 各产品线保持清晰定位，并在统一生态内相互配合；
- 开放接入： 支持不同类型的用户、节点和资源提供者参与网络；
- 资源共享： 提高全球分散带宽与算力资源的利用效率；
- 统一激励： 通过 BTT 连接资源使用、服务支付和节点激励。

各产品线通过统一的技术体系与经济机制形成协同，共同为 Web3 与 AI 应用提供开放、高效且可持续的基础设施支持。

3. 生态架构

BitTorrent 作为一级主品牌，统一承载生态的技术认知、用户基础和品牌价值；BitTorrent、BTTInferGrid 和 BTTC 分别承担数据与流量分发、去中心化算力服务、跨链通信与价值结算等职责，BTT 则用于连接各类资源的使用、支付、激励、质押和治理。

各产品线既可以独立向用户和开发者提供服务，也能够通过标准化接口和 BTT 经济机制协同运行，形成“数据分发—算力调用—结果验证—链上结算”的完整服务链路，为 Web3 与 AI 应用提供去中心化基础设施支持。

3.1 分发层：BitTorrent

BitTorrent 是生态的流量入口与数据分发层。依托成熟的 P2P 协议、客户端体系和分布式节点网络，BitTorrent 将用户终端的闲置带宽组织为可共享的分发资源，为文件和数据提供点对点传输能力。

在传统文件分发场景中，BitTorrent 通过种子、磁力链接、节点发现、文件分片和智能带宽调度等机制，提高数据传输效率并降低对中心化服务器的依赖。

面向 AI 场景，BitTorrent 的分发能力将进一步用于开源模型权重、训练数据集及其他大规模数据资源的传输，为 BTTInferGrid 的推理计算和存储服务提供模型与数据入口。

3.2 去中心化算力层：BTTInferGrid

BTTInferGrid 是面向 AI 推理场景的去中心化算力网络，连接全球分布的 GPU 节点与算力需求，为开发者和应用提供弹性、低成本、可验证的推理计算服务。

开发者通过统一接口提交推理请求，网络根据任务需求、节点性能和资源状态完成算力匹配与任务调度。符合要求的 GPU 节点参与任务执行，并通过贡献算力获得收益。

BTTInferGrid 通过任务挑战、结果比对、节点评分和链上协调等机制验证计算结果，并将验证结果作为费用结算、奖励分配和违规惩罚的依据。

3.3 网络与结算层：BTTC

BTTC 是生态的跨链互操作、Layer 2 扩容与链上结算层。它连接 TRON、Ethereum、BNB Chain 等异构区块链网络，为资产、数据和服务状态的跨链流转提供基础设施。

BTTC 通过合约层、Delivery 链和 BTTC 链构成的三层架构处理跨链请求。合约层负责资产锁定、释放和事件记录；Delivery 链负责监听跨链事件、生成检查点并组织验证人签名；BTTC 链负责智能合约执行、状态更新和应用运行。凭借 EVM 兼容、较低 Gas 成本和高频交易处理能力，BTTC 可以承载生态中的服务结算、节点质押、奖励分配和跨链支付。

4. 产品一：BitTorrent 生态的“流量与分发引擎”

BitTorrent 生态的核心价值，在于把分散在海量用户终端上的上传带宽、存储空间与内容副本，组织为一个无需中心协调者的协同分发网络。作为运行超过二十年、部署规模最大的去中心化内容分发协议，BitTorrent 在本生态中承担流量与分发引擎的角色：它是所有大体积数字资产——文件、软件更新、模型权重、数据集——的数据平面，是生态节点与钱包的规模入口，也是"资源贡献—代币获取—服务消费"这一经济闭环在协议层的起点。

4.1 设计原则

4.1 设计原则

在 BitTorrent 生态设计中，我们遵循以下五项原则：

- **开放与兼容：**采用协议扩展（extension message）实现功能升级，确保新旧客户端“软分叉”共存，避免破坏现有网络。
- **用户主权与自愿参与：**资源贡献完全透明且由用户自主配置。协议严控资源使用边界（带宽与存储），绝不占用前台业务资源，用户可随时退出。
- **端到端可验证：**不依赖节点信任。通过密码学签名和逐分片验证，确保数据完整性与经济交换的安全性，有效控制风险。
- **协议中立与市场驱动：**协议仅提供计量与结算原语，不干涉内容价值裁决。内容分发与存储价值完全由市场供需决定。
- **用户友好：**降低准入门槛，将钱包功能内置于客户端并自动化出价，让普通用户无需理解区块链复杂概念即可便捷参与资源共享。

基于此，分发引擎由协议层（寻址与交换）、客户端层（覆盖主流终端）、激励层（Speed 带宽撮合）与数据能力层（支持 AI 模型与数据集）共同构成。

4.2 P2P 分发协议与网络原语

4.2.1 内容寻址与完整性验证

分发对象在发布时被切分为固定大小的分片（piece），并生成描述分片结构与哈希的元数据（metainfo）。协议以元数据的加密摘要——infohash——作为分发对象的全局唯一标识符。寻址的是内容本身而非其存放位置：只要 infohash 一致，数据来自哪个节点、经过何种路径，均不影响其正确性。

这一设计给出两条基本保证。

第一，**完整性是端到端且逐分片可验证的**。任何节点收到的任何分片，都可独立对照元数据中的分片哈希验证；传输损坏与错误分片在分片粒度即被发现并丢弃，整个验证过程不依赖对任何中间节点的信任。我们同时明确该保证的边界：分片哈希证明的是字节完整性——收到的内容与元数据所描述的内容一致；发布者身份与内容可信性由签名等上层机制建立（见 4.5.2 节的双重校验设计）。

第二，**数据来源无关（source-agnostic）**。同一 infohash 的数据可同时来自 P2P 对等节点与 HTTP 源（Web Seed，BEP-19），两类来源受完全一致的完整性约束。P2P 分发因此可以与既有中心化基础设施——对象存储、CDN、镜像站——混合部署：HTTP 源保证基线可用性，P2P 层承担峰值卸载，内容提供方无需改变现有存储架构即可渐进接入。

BitTorrent v2（BEP-52）将哈希算法升级为 SHA-256，并为分发对象中的每个文件构建独立的 Merkle 哈希树，由此获得文件粒度的内容寻址与跨种子去重、以 Merkle 根为锚点的细粒度校验、以及按文件选择性获取的能力。

4.2.2 对等发现与组网

围绕同一 infohash 交换分片的节点集合构成群（swarm）。节点通过三种互补机制发现彼此：

- **Tracker**（BEP-3/BEP-23）：节点向 Tracker 声明（announce）自身持有的内容，Tracker 返回同一 swarm 内的其他节点，是由发布方或第三方运营的集中式介绍服务；
- **DHT**（BEP-5）：基于 Kademlia 的分布式哈希表，将"infohash → 节点列表"的映射分散存储于全网节点，无需任何集中式设施即可完成查询；
- **PEX**（BEP-11）：已连接的节点之间直接交换各自已知的对端列表，加速 swarm 内的连通收敛。

DHT 与 PEX 使 swarm 在没有任何中心化协调者的情形下仍可组网并维持连通；Tracker 退化为可选的加速与统计设施。发现层不存在单点：任一机制失效或被移除，swarm 均可经由其余机制继续运转。

4.2.3 分片交换与带宽分配

swarm 内的资源分配由两个算法共同决定。

稀有分片优先（rarest-first）。 节点在选择下一个请求的分片时，优先请求当前 swarm 中副本数最少的分片。该策略使分片分布趋于均匀，降低关键分片因少数持有节点离线而消失的概率。swarm 的健康程度可以用可用性指标（availability）度量，通常表示为群内分布式副本的数量：只要每个分片都至少有一个活跃节点持有，文件即处于"可用"状态，即使群内暂时没有完整种子。

choking 与 tit-for-tat 撮合。 每个节点仅向有限个对端开放上传。客户端将对端分类为 unchoked（可接收数据）与 choked（暂不服务）两种状态：一个典型的 choking 算法按"自上一轮以来对方向我上传的数据量"对对端排序，前 n 名（n 为该客户端设定的 unchoke 槽位数）获得 unchoke，其余被 choke。choke 状态周期性重算，典型周期为 15 秒。种子节点不再需要下载，因此以"向各对端发送的数据量"为排序依据，单纯优化整体吞吐。

在常规槽位之外，客户端保留少量**乐观解锁（optimistic unchoke）**槽位（典型为一个），以随机或轮转方式选择对端予以 unchoke。这为尚未建立互惠关系的新节点提供了获取初始分片的机会，使其得以启动与其他节点的互惠交换。

choking 是 swarm 内资源分配的主通道：贡献上传越多的节点获得越好的下载服务，最低产的连接被降级、断开或封禁，搭便车行为在协议层被自然抑制。这一机制同时是 4.4 节将 BTT 出价并入分配决策的挂载点。

4.2.4 传输与拥塞控制

数据传输默认运行于 uTP（BEP-29）之上。uTP 采用 LEDBAT 拥塞控制：以单向时延变化为拥塞信号，在检测到链路排队时主动降速，让位于同一链路上的网页浏览、视频会议等交互式流量——包括其他设备上的应用对同一出口链路的使用。对分发网络而言，这一特性具有结构性意义：后台分发任务可以长时间利用闲置上行带宽而不劣化前台体验，消费级设备因此可以长期在线、持续贡献。

4.3 客户端节点网络

分发引擎的执行层，是 BitTorrent 与 µTorrent 两个客户端品牌构成的产品矩阵，以及实现相同协议族的众多第三方客户端组成的开放网络。开放协议保证了客户端市场的多样性与竞争；官方客户端矩阵覆盖主要终端场景（表 4-2），累计下载量超过 20 亿次。

表 4-2：BitTorrent 官方客户端矩阵

客户端	定位	在分发生态中的作用
BitTorrent Web	边下载边播放，面向普通用户	降低内容消费门槛，将播放与 P2P 传输结合
BitTorrent Classic	桌面批量下载与高级配置	为批量分发、做种、任务优先级与带宽调度提供细粒度控制
BitTorrent Android	移动端 Torrent/Magnet 下载与播放	将分发网络延伸至移动场景，支持 Wi-Fi-only 与远程添加桌面任务
BitTorrent Neo	macOS 现代轻量客户端	覆盖新版 macOS 的轻量下载体验

BitTorrent Remote	远程访问家庭客户端	跨设备任务控制入口

对生态而言，客户端网络承担三个结构性角色。

节点密度与地理分布。 swarm 的吞吐与韧性直接取决于节点的数量与分布。大规模安装基数意味着任意热点内容都可能在没有预置基础设施的地区获得本地化的高吞吐分发。客户端不仅是内容的消费端：在用户允许的范围内，每个客户端都持续向 swarm 上传已有分片，使消费节点同时成为分发节点——这正是表 4-1 中规模正反馈的微观来源。

钱包与身份的自然载体。 BTT 钱包内置于支持的客户端，随客户端版本自然分发到大众用户手中。身份模型延续协议的既有约定：BitTorrent 协议自诞生起标识的就是软件实例——IP 与端口意义上的客户端——而非自然人。钱包与客户端实例绑定沿用同一思路，与加密代币"谁持有密钥即谁的代币"的身份框架一致；除钱包口令外，我们不引入额外的身份管理层。用户无需理解密钥管理即可参与 BTT 的获取与支出（原则五）。

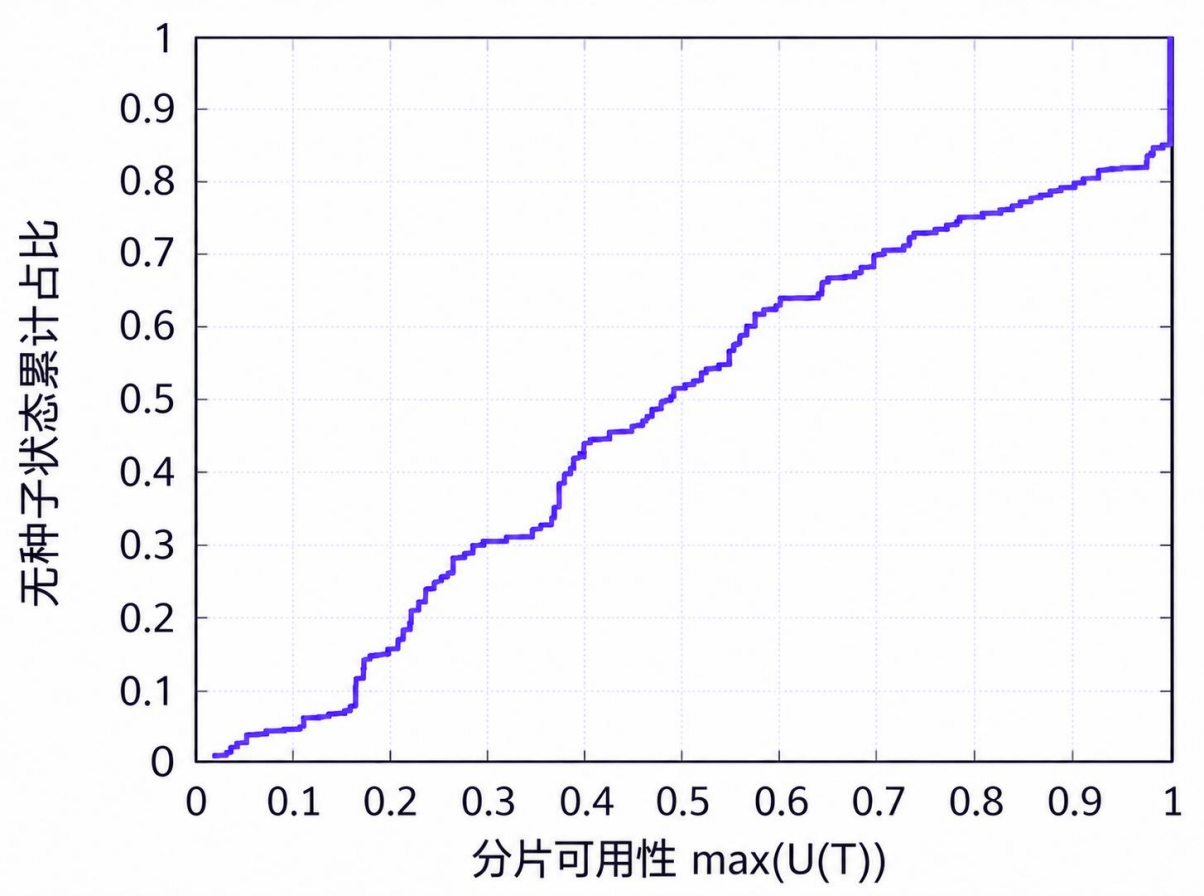
供给侧入口。 具备长在线时间、闲置上行带宽与本地存储的客户端节点，构成 4.4 节带宽激励机制的供给方，也是生态内其他资源网络扩展节点时的天然候选集合。

4.4 BitTorrent Speed：基于 BTT 的带宽激励层

4.4.1 问题定义：做种激励的结构性缺失

原生协议的 tit-for-tat 互惠仅在下载进行期间成立。由于上传与下载速度的普遍不对称，下载方通常在上传等量数据之前就已完成整个文件的下载；而文件一旦下载完成，继续上传（做种，seeding）在协议内没有任何回报，用户关闭客户端不承担任何代价——做种是纯粹的利他行为。其系统性后果是：用户离开 swarm 时上传量普遍少于下载量，swarm 的存续时间短于内容的需求周期。

rarest-first 与可用性机制可以在无种子状态下延缓衰退——只要每个分片仍有节点持有，下载仍可完成。但对无种子 swarm 的大规模实证测量表明：约 86% 的无种子状态无法依靠群内存量节点的分片集合重建完整文件，能够完成集体重建的仅约 14%（Kaune et al., 2010）。



做种激励缺失是协议层的结构性缺陷。我们曾系统评估过在协议核心内置长期激励的路线，并明确否决：其一，"什么内容值得长期做种"无法在协议层清晰定义而不被博弈利用；其二，BitTorrent 生态之大使任何硬分叉几乎必然失败——不做种即惩罚的协议，用户只需换用不惩罚的旧客户端即可绕过；其三，与其把价值判断包进协议，不如把判断权交给市场。因此我们的答案是：保持协议核心不变，在其上以扩展形式引入代币化的带宽交换——BitTorrent Speed。这是原则一与原则四的直接推论。

4.4.2 总体设计

- BitTorrent Speed 是叠加在现有协议之上的激励覆盖层，自 2019 年起集成于 BitTorrent 与 µTorrent 客户端运行。它不改变协议的任何既有功能：不参与 BTT 交易的用户，原有 Torrent 功能不受任何影响；在新旧客户端混合的 swarm 中，旧客户端行为保持不变，仅在支持 Speed 的客户端之间叠加代币交换，旧客户端依旧按原生互惠逻辑最大化上传。

在 Speed 中，节点按行为分为**服务请求方**（service requester，以 BTT 换取对端带宽）与**服务提供方**（service provider，以带宽换取 BTT），同一节点可同时扮演两种角色。生命周期从既有协

议的对等发现机制开始——Speed 不引入新的发现基础设施，swarm 的组建与 4.2.2 节完全一致。BTT 钱包、自动出价与收益看板内置于客户端；该机制不是挖矿——收益来自真实的带宽服务交换，而非算力竞争。

关于加速效果，我们采用严谨的表述：Speed 不创造物理带宽，也不保证任何单次下载变快；实际效果取决于 swarm 内的做种数量、对端可用带宽、网络连接质量、文件需求热度与参与客户端的占比。其机制本质是**通过 BTT 出价获取优先供给，从而提升下载速度的可能性**——将稀缺的优质上传能力经济地引导给需求最高的节点，同时给予做种者留在 swarm 中的经济理由，延长文件的可用周期。

4.4.3 竞价机制

出价广播。 服务请求方通过协议扩展消息，向每一个持有其所需分片的对端发送出价，消息内容为请求方愿意为每个分片支付的 BTT 数量。

自动出价。 出价默认由客户端代表用户自动完成。初始版本采用简化的自动出价算法：

bid =（spending rate × 钱包内剩余 BTT 余额）/ 剩余下载量（KB）

其中 spending rate 为出价激进程度参数，取值 0.0–1.0，初始版本设为 1.0；剩余下载量为零时自动停止出价。该公式意味着出价随下载进程动态变化——由于对端时常免费提供数据（原生互惠仍在运行），以"剩余预算 ÷ 剩余数据量"计算的出价会随下载推进而自然上调。

出价修订与频率控制。 客户端可以采用任意启发式决定何时重新广播新出价，但出价变动不超过上一次出价 10% 时不重新广播，且重新出价频率不高于每分钟一次，以控制协议消息开销；用户手动修改出价金额时立即广播。算法的后续演进方向包括：基于 swarm 内出价消息流量与当前传输状况估计带宽的市场价格，并结合分片稀缺度信息调整出价。

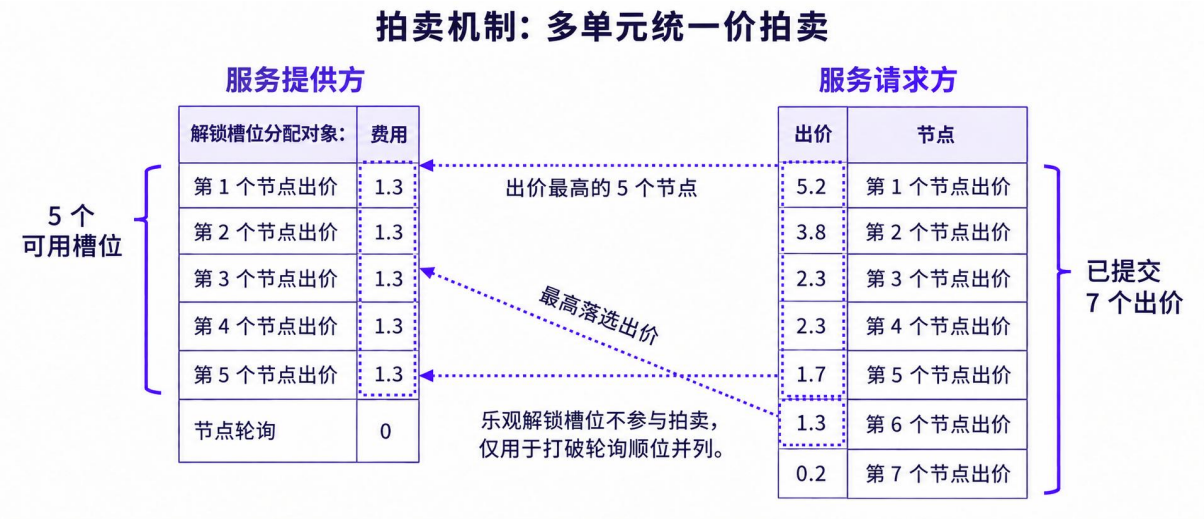
用户控制。 我们为用户保留对出价过程的完全控制：整体开关、按种子开关、支出速率调节、保留价设置，以及对出价过程的细粒度干预（原则二）。

4.4.4 撮合与拍卖

服务提供方将 BTT 出价并入 4.2.3 节的 choking 决策。到达 choking 重算周期时，提供方首先汇总当前有效出价，将既有排序算法修改为按"最高有效出价 + 历史数据互惠量"的组合排序分配

unchoke 槽位。中标者被 unchoke 时，提供方先以协议扩展的出价响应消息告知其应支付的费率（BTT/byte），随后发送常规 unchoke 消息——扩展消息与原生消息在此严格分层，互不干扰。

客户端可以实现任意拍卖格式。我们推荐 Vickrey–Clarke–Groves 机制的变体，其在此场景下收敛为**多单元统一价拍卖（multi-unit uniform price auction）**：每个请求方对每个提供方只竞争一个 unchoke 槽位，全部中标者按**最高落选出价**统一计费。统一价与次价计费的组合抑制了策略性报价——如实出价即是近似占优策略，请求方无需博弈他人报价。



乐观解锁的保护。 乐观解锁槽位承担着让新节点进入 swarm 的引导职能，必须谨慎对待：它不参与常规拍卖，仅当多个从未被 unchoke 过的节点在轮转顺位上完全并列时，才允许以拍卖破除平局。这保证了没有 BTT 的新参与者永远保有免费进入网络的路径——激励机制不得反噬网络的开放性（原则四与原则五的交汇点）。

4.4.5 支付通道与结算

支付在请求方与提供方之间的点对点支付通道中完成，通道运行于高吞吐结算账本之上，与 BTT 主账本双向衔接。协议如下（K1 为请求方公钥，K2 为提供方公钥）：

1. 请求方生成公钥 K1，并向提供方索取其公钥 K2；
2. 请求方在结算账本上开立以 K2 为收款方的支付通道，并将一笔 BTT 从 K1 账户转入通道作为托管余额；
3. 请求方生成一笔通道内交易：将首期应付金额分配给 K2，通道内剩余 BTT 归还 K1；
4. 请求方对该交易签名，将交易与签名一并发送给提供方；

5. 提供方首次见到该通道时，向结算账本查询通道信息，核对与请求方所发内容一致后方才认可该笔支付；
6. 此后每完成一个分片的交付，请求方调整交易、将更多金额分配给 K2，重新签名后发送——支付随交付逐分片递增；
7. 通道托管余额耗尽而交付仍需继续时，请求方开立新通道，重复步骤 2–4。

传输完成后，提供方对其见到的最后一笔交易签名并提交结算账本，按最终分配额清算，通道随之关闭、不可复用。若提供方中途停止而使通道资金悬置，通道内置的超时机制生效：超时到期时账本自动关闭通道，并将通道内全部 BTT 退还请求方。

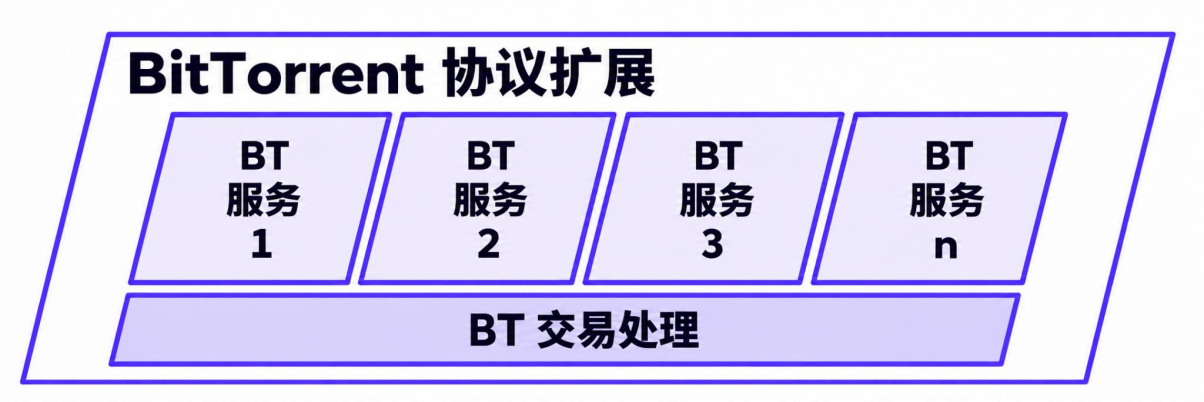
失败与惩戒。 BTT 转账在超时后仍未完成的请求方将被 choke、不再获得数据；反复拒付者被提供方封禁——断开连接且在设定期限内拒绝其重连。对称地，交付无效数据的提供方同样面临封禁。

风险敞口分析。 双方逐步交替推进——每一步交付一个分片或递增一笔签名金额——因此任意时刻：提供方的最大损失为一个分片的带宽（先交付、后收款）；请求方的损失为零（分片经完整性验证后才递增支付）。风险敞口在协议上有界且不对称地偏向保护贡献资源的一方，这是原则三在经济层的落实。

4.4.6 从带宽到通用资源交换

Speed 的多年运行验证了一个基础判断：**以代币在小额、高频、互不信任的对端之间计量并交换基础设施资源，在消费级大规模客户端网络中是工程可行的**。参与门槛（内置钱包、自动出价）、资源保护（配额与拥塞控制）与激励相容（做种收益）可以在同一个大众产品中并存。

带宽交换所依赖的协议扩展与交易处理是通用的：同一套"出价—撮合—通道支付"原语可以承载更多资源类别的服务。我们将协议扩展设计为可插拔的服务框架——多个 BT 服务并列运行于统一的交易处理层之上，新服务随需求出现被定义、评审并加入网络，其标准化沿用 BEP 的开放流程。



4.5 面向模型与数据集的分发能力

4.5.1 负载特征与适配性

AI 工作负载为内容分发引入了一类新的对象：模型权重、训练与评测数据集、训练检查点、容器镜像与配套资源包。BitTorrent 协议对文件内容类型无感，其通用大文件分发能力可以自然映射到这类资产；而这类资产的负载特征，恰好落在 4.2 节协议属性的最优适用域内：

- **单对象体积大**：主流开源模型权重从数 GB 至数百 GB，数据集常以 TB 计——正是分片化多源并行下载的目标场景；
- **发布时刻全球并发拉取**：新模型版本发布呈"惊群"模式，数小时内全球并发下载——正是 swarm 规模正反馈的发力点，需求峰值直接转化为供给峰值；
- **只读、不可变、热点集中**：权重与数据集发布后不原地修改——与内容寻址的不可变语义天然一致；
- **完整性敏感**：权重被篡改即构成模型供应链攻击——逐分片哈希验证提供字节层的第一道防线；
- **长尾版本需长期可取**：科研复现与生产回滚要求历史版本持续可用——这是做种问题在 AI 负载上的重现，也是 Speed 激励原语的用武之地。

与之相对，中心化对象存储与 CDN 在该负载下同时面临出口带宽成本随下载量线性增长与发布峰值拥塞两个问题。表 4-3 给出底层能力到 AI 资产分发价值的映射，同时明确标出每项能力之上应由上层系统补足的部分。

表 4-3：分发层能力对 AI 资产分发的映射与边界

底层能力	对模型/数据分发的价值	上层系统补足的能力
大文件分片、多源并行下载	分发大体积权重、数据集、检查点、容器镜像与资源包	资产目录、版本号、依赖管理、差分更新
分片哈希验证	发现传输中的数据损坏与错误分片	发布者签名、信任链、SBOM/模型卡、来源证明
swarm 内副本复用	热门资产的下载者同时成为供给者，分担源站出站带宽	企业网络策略、可观测性、调度与限流
BTT 带宽激励	对长期做种与优先供给建立经济激励	价格、预算、防作弊、结算与合规策略
BTFS 内容寻址存储	为资产提供存储、共享与按 CID 访问的基础	私有数据加密、访问控制、删除/撤回、数据主权与合规

4.5.2 工件分发约定

为使上述能力可被工程化使用，我们在 BitTorrent v2 元数据能力之上，将模型与数据集的标准分发单元约定为**工件种子（Artifact Torrent）**。该约定包含三项内容。

不可变寻址。 每个模型或数据集版本对应唯一的 infohash：版本即内容，不存在原地覆盖。消费方引用某一版本时引用的是其内容摘要，而非可被重定向的 URL。基于 v2 的 per-file Merkle 根，权重文件可与发布方签名一同构成校验凭据：Merkle 根验证字节完整性，签名验证发布者身份——二者共同构成供应链校验，缺一不可。消费方在任意来源（P2P 对端、Web Seed、镜像）下载后均可离线完成全部验证。

结构化元数据。 模型卡或数据集卡、许可证标识、版本谱系（父版本 infohash）作为结构化元数据随工件种子分发。许可信息与内容不可分离地绑定：消费方在获取资产的同时，必然获取其使用条款与来源谱系。

文件粒度按需获取。 基于 v2 的按文件寻址，消费方可以仅拉取工件中所需的部分——例如特定量化精度的权重分卷、数据集的特定子集——而无需下载完整工件。

4.5.3 分发模式

工件分发支持三种可组合的部署形态。

公开 swarm 面向开源模型与公开数据集的发布分发。发布方仅需维持初始种子与一个 Web Seed 源；版本发布后的峰值带宽由 swarm 聚合承担——下载热度越高，聚合供给越大，发布方源站的出站带宽压力反而趋于平缓。

私有 swarm 面向企业与联盟场景。swarm 准入受控（节点须持有凭证方可加入），可与内网、专线环境共存，用于组织内大规模节点的模型同步——例如推理集群的权重滚动更新、边缘节点的批量镜像分发。

混合源分发 面向渐进接入。Web Seed 使既有模型仓库与对象存储可以原样作为数据源参与 swarm，P2P 层作为带宽卸载与加速层叠加其上；发布方无需改变现有存储架构，即可按流量比例逐步把出站带宽转移到 P2P 层。

4.5.4 长尾可用性与激励

模型与数据集的长尾版本存在"低频但必须可取"的可用性需求：科研复现需要引用论文所用的确切版本，生产系统需要可回滚的历史检查点。在纯利他做种下，这类低热度工件正是最先从网络中消失的部分——4.4.1 节的做种问题在 AI 负载上原样重现。

我们对此复用 Speed 的激励原语，而不引入任何中心化的价值判断：需要某一版本长期可用的一方——发布方、依赖方或任意第三方——以 BTT 持续出价，做种节点据此获得留存该工件、维持其在线的经济理由。哪些工件值得留存，由市场出清决定，而非由协议或运营方裁决（原则四）。

4.5.5 定位边界

我们将 BitTorrent 生态在 AI 基础设施中的位置界定为：**模型与数据集的分发数据面（distribution data plane）**——而非模型注册表、数据治理平台或推理调度网络。

这一界定基于能力的诚实边界。分片哈希验证字节是否传输正确，但不单独证明模型由谁发布、是否安全、是否可合法使用；分发层也不自动解决版本治理、许可执行、数据隐私与撤回问题。面向企业或受控 AI 场景，应在 BitTorrent/BTFS 之上叠加发布者签名与信任链、模型卡与数据卡、版本目录、访问控制、加密与合规审计（表 4-3 右列）。这些上层能力由具体应用与第三方系统实现，分发层保持中立与通用——一个通用、可验证、被亿级设备实现的分发层，正因为不试图替代其上的治理系统，才能被所有治理系统放心复用。

对去中心化算力网络而言，分发层承担"模型到节点的最后一公里"：计算节点在承接任务前，经工件种子从最近的 swarm 拉取模型镜像，完成 Merkle 校验与签名验证后加载执行；冷启动带宽成本由 P2P 层承担，同一模型在某一区域的任务热度自动转化为该区域内更高的分发吞吐。这种分层架构确保了 BitTorrent 分发层不仅是一个协议，它是整个 BTT 生态中承载 AI 数据流动的血液，保证了资源（算力/存储）与价值（代币）在底层的顺畅连接。

5. 产品二：BTTInferGrid 生态的“去中心化算力网络”

BTTInferGrid 是 BitTorrent 生态面向 AI 推理的分布式计算网络，通过连接参与者贡献的 GPU，将链下任务执行、结果验证与链上协调和结算结合起来。网络以真实推理需求驱动资源供给，目标是降低开发者获取算力的门槛，并让闲置 GPU 在满足性能、可靠性和验证要求的前提下转化为可持续服务能力。

5.1 产品定位：生态的计算与存储层

BTTInferGrid 聚焦高频、可并行的 AI 推理工作负载，为开发者和应用提供弹性调用的算力资源。相较由单一云服务商统一运营的模式，网络通过分布式 GPU 供给扩大资源来源，并以任务调度、挑战验证、评分共识和经济约束管理服务质量。其架构由硬件资源、链下执行、验证以及链上协调四个部分组成：GPU 节点执行任务，验证节点评估结果，链上合约记录评分、质押、罚没和奖励。

存储能力用于支持模型权重、输入数据、结果摘要及任务凭证的管理，并与 BitTorrent 的大文件分发能力协同。BTTInferGrid 的核心边界是组织计算资源和验证服务结果，而不是承诺所有推理结果都能由区块链直接证明；不同模型和任务需要采用与成本、确定性和安全要求相匹配的验证方式。

5.2 参与角色与资源接入机制

网络包括需求方、算力节点、验证节点和数据服务节点。需求方通过 API 或应用提交推理请求并支付费用；算力节点贡献 GPU，在本地执行任务并返回结果；验证节点负责节点选择、挑战任务、结果评分和共识聚合；数据服务节点在相关能力上线后，可承担模型与任务数据的存储、可用性和检索服务。

算力节点接入时需注册设备并通过硬件性能、网络连接、可用性和基准任务检查。调度系统持续依据延迟、任务完成率、挑战准确率和历史可靠性更新节点评分。早期网络可通过性能门槛和白名单控制供给质量，待验证和治理机制成熟后再逐步开放。验证节点需满足准确性与可用性要求并质押 BTT；退出、解绑和质押释放应经过等待期，未完成任务、未结争议或待执行处罚处理完成后方可退出。

5.3 AI 推理任务与算力调度流程

推理服务形成从请求到结算的操作闭环：

1. 任务提交：用户通过 API 或应用接口提交模型、输入、性能要求、费用上限和超时条件。
2. 资源匹配：验证节点或调度模块根据 GPU 能力、节点评分、延迟、可用性和任务价格选择算力节点。
3. 链下执行：算力节点获取必要的模型与输入，在本地完成推理，并返回输出、执行元数据和结果摘要。
4. 结果评分：验证节点通过抽样复核、挑战任务或多节点比较评估结果，聚合评分并过滤异常值。
5. 状态记录与结算：最终评分和任务状态写入链上，合约依据已验证工作量、服务质量和计费规则分配费用与奖励。

节点超时、拒绝任务或返回无效结果时，调度模块可将任务重新分配给其他节点，并在评分中记录失败。是否重试、重试次数和费用承担由任务策略确定；需要高可用性的应用可选择冗余执行或更高服务等级，但会承担相应的额外成本。

5.4 可验证推理与服务质量机制

链下推理存在结果错误、跳过计算、复用缓存输出、验证者偏袒和合谋等风险。BTTInferGrid 采用“冗余评估、激励对齐、渐进验证”的思路：验证节点将难以与真实请求区分的挑战任务随机注入工作负载，根据预期的统计或语义属性评估响应；同一算力节点由多个验证节点独立评分，再通过 Yuma 共识聚合并过滤极端值，降低单点操纵和协同抬分的影响。

节点评分综合挑战准确率、任务完成量、延迟、可用性和历史可靠性。算力节点奖励与共识评分及已验证工作量挂钩；验证节点的评分若持续偏离共识，其声誉权重和奖励会下降，严重异常时触发质押罚没。验证机制可从统计共识和挑战审查逐步扩展到可信执行环境（TEE）、确定性重执行和零知识证明（ZK），但应明确不同方案的适用范围和成本。对于具有随机性的生成式模型，验证结果通常表示达到协议定义的可信度，而非数学意义上的绝对正确。

5.5 分布式存储与数据服务能力

分布式存储作为与推理协同的扩展能力，用于保存或分发模型权重、输入数据、执行日志、结果摘要和验证凭证，减少重复传输并提高任务数据的可用性。大文件可通过 BitTorrent 网络进行分发，BTTInferGrid 负责将可验证的文件标识、版本信息和访问条件关联到具体任务。

存储服务需要明确客户端加密、文件分片、冗余策略、完整性校验、检索证明、网关权限和数据删除规则。涉及敏感数据时，应遵循最小披露原则，不将原始输入和私密结果直接公开上链；链上仅记录任务状态、哈希承诺和结算所需信息。相关能力应在完成技术规范、性能测试和安全审计后逐步开放。

5.6 BTT 结算、节点激励与惩罚机制

BTT 用于支付推理及后续数据服务费用，也可作为验证节点质押和违规罚没资产。用户费用可以按请求次数、计算时长、吞吐量、模型规格和服务等级计价；价格根据可用 GPU 供给、推理需求和节点质量动态调整，并可设置成本下限，避免节点在无法覆盖硬件、电力和带宽成本时持续提供低质量服务。

算力节点收益由已验证工作量、共识评分、任务完成率和历史可靠性共同决定，闲置资源或未验证工作不获得有意义的奖励。验证节点因执行挑战、评分和识别异常获得奖励；偏离共识、操纵评分或参与合谋时，其奖励降低并可能被部分或全部罚没。协议费用可用于网络运营、奖励池或按既定规则处理。长期激励应逐步从补贴和代币释放转向真实用户费用，供给扩展与网络利用率挂钩，降低无需求扩张和奖励稀释风险。

5.7 开发者接口与应用接入

开发者通过统一 API 提交任务、查询状态、获取结果和查看费用，可按模型、延迟、预算、验证等级及冗余策略声明服务要求。接口应返回任务标识、节点选择状态、计费明细、验证结果和错误原因，并支持超时、取消、重试和异步回调。SDK 可封装鉴权、模型参数、文件标识、支付和结果解析，降低现有 AI 应用接入成本。

面向生产环境还需提供测试环境、配额管理、价格查询、节点与任务监控、服务等级指标和开发者支持渠道。参考白皮书仅明确了 API/应用接口及企业专用计算池、SLA 服务等方向，具体 SDK、测试环境和监控能力仍需以正式产品规范为准，白皮书不将其表述为已经上线的功能。

6. 产品三：BTTC 链上执行与生态结算网络

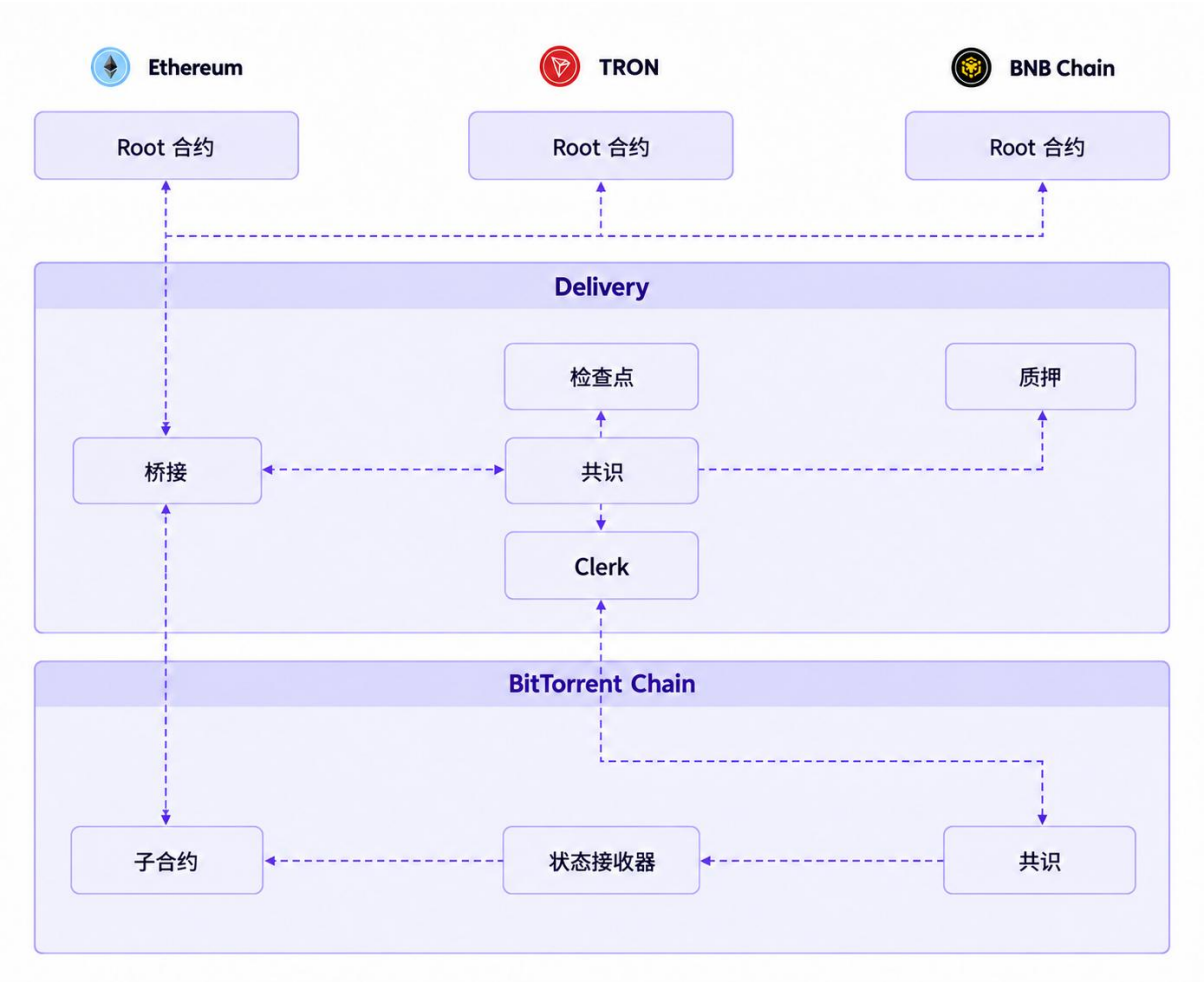
BTTC 是 BitTorrent 生态的链上执行与结算基础设施，为算力、存储、分发及其他生态服务提供统一的合约运行、状态记录和价值结算能力。通过 EVM 兼容执行环境、验证人网络和 BTT 经济机制，BTTC 将服务请求、业务状态与结算结果组织为公开、可验证、可审计的链上流程。

6.1 产品定位、用户与场景

BTTC 面向四类主要参与者。普通用户可以低成本使用链上应用并以 BTT 完成支付；内容发布者、数据与模型提供方可以登记服务、授权关系和结算状态；开发者可以使用 EVM 兼容环境部署 DeFi、链游、支付及生态服务合约；算力与存储服务提供方则可以通过 BTTC 完成任务确认、服务计费、质押约束和收益结算。

BTTC 的核心价值是为生态服务建立可组合的可信协调层。BTTInferGrid、BitTorrent 分发网络或第三方应用可以将服务请求、节点状态和结算凭证提交至 BTTC，由智能合约按照既定规则完成状态确认、费用扣除、收益分配和争议记录。对于高频应用，BTTC 提供低成本的合约执行环境；对于资源服务，BTTC 提供统一的任务登记、证明提交和链上结算接口。

6.2 架构：三大功能层、关键参与角色



BTTC 采用三层模块化架构，各层职责相互独立，并通过链上事件和共识机制协同工作：

- 合约层（Root Contracts）：承担网络治理和资产管理入口，负责验证人注册、质押与委托、参数配置、费用管理以及生态服务合约的状态记录。
- Delivery 链：承担验证人协调、共识确认和网络治理状态维护。验证人对网络事件和状态变更进行校验，通过 PoS/BFT 共识形成一致结果，为 BTTC 链运行提供稳定的验证与治理基础。

- BTTC 链：承担智能合约执行和业务状态承接，兼容 EVM 与 Solidity，负责处理交易、执行应用逻辑、记录服务状态并完成费用与收益结算。

网络中的关键角色包括验证人、区块生产者、委托者和开发者。验证人参与 Delivery 链共识及网络治理；区块生产者从验证人中轮值产生 BTTC 链区块，负责交易打包与状态执行；委托者通过质押或委托参与网络并分享激励；开发者部署应用、服务协议和结算合约。各角色的权限、质押和操作记录均由链上规则约束。

6.3 机制：服务执行与链上结算流程

BTTC 将生态服务的请求、执行确认和费用结算组织为标准化链上流程，主要包括五个环节：

1. 登记请求：用户或应用向服务合约提交任务、资源使用或支付请求，合约记录请求参数、费用上限和唯一标识。
2. 确认服务：算力、存储或分发节点接受任务，系统记录服务方、执行条件和预计交付状态。
3. 提交结果：服务完成后，节点或验证模块提交结果摘要、完成证明及必要的质量指标。
4. 规则校验：智能合约按照协议设定的状态、签名、时限和计费规则检查提交结果，拒绝重复、超时或格式不合规的请求。
5. 执行结算：验证通过后，合约使用 BTT 完成费用扣除、节点收益分配、协议费用处理和状态归档；发生异常时，则进入重试、扣罚或争议处理流程。

该流程将服务交付与资金结算连接起来，使请求、执行者、结果摘要和支付记录均可追踪。不同生态服务可以复用统一的账户、Gas、质押和结算能力，同时根据业务需要定义任务格式、服务质量、确认条件与争议规则。BTTC 只负责链上协调和结算，不替代算力执行、文件传输或存储检索等链下服务本身。

6.4 机制：共识机制与安全模型

BTTC 的安全性建立在 Delivery 链共识与 BTTC 链出块机制的分工之上。Delivery 链采用基于 Tendermint 的 PoS/BFT 共识，每轮由提议者提出区块或状态更新，验证人依次完成 Pre-vote 和 Pre-commit。超过三分之二验证人签名后，状态才被视为最终结果；当节点故障或作恶超过三分之一时，共识停止而不是确认错误状态，从而保护网络状态的一致性。

BTTC 链采用验证人轮转的区块生产机制。区块生产者负责打包交易、执行智能合约和更新状态，轮转机制提升了吞吐和出块可预期性；连续漏块的生产者可以被替换，以降低单个节点故障对网络性能的影响。验证人的签名密钥与控制权限分离，关键状态更新必须满足阈值签名要求，并由链上合约按照协议规则进行验证。

安全模型覆盖服务登记、状态更新和结算执行的全过程：每个请求使用唯一标识防止重复处理；链高、时间戳和状态机约束用于限制重放与越序执行；关键状态必须经过共识确认，不能由单一验证人独立修改；合约只接受符合格式、权限和签名规则的输入。验证人质押、签名出勤、出块表现及委托关系均通过链上机制记录，为违规扣罚、节点替换和社区监督提供依据。

7. BTT 经济模型与代币设计

BTTC 的经济模型旨在构建一个自驱动、可持续、抗中心化滥用的激励体系。通过围绕原生代币 BTT 的质押机制、手续费体系、委托逻辑与治理工具，BTTC 保障网络的安全性、验证人活跃度以及生态增长动能，为去中心化跨链架构提供长期动力支持。

7.1 核心代币：BTT

BTTC 网络以 BitTorrent Token（BTT）为核心资产，兼具质押、激励和链上支付功能。验证节点需质押 BTT 参与共识。此外，跨链操作的手续费和激励均使用 BTT 支付。同时，链上交互、智能合约部署及数据存储也以 BTT 作为交易媒介。BTT 发行量固定，通过链上智能合约严格管控，以确保通胀的可控性和经济活动的透明性。

BTT 原始发行总量为 990,000,000,000 枚。BTTC 主网上线并实施 BTT redenomination 后，旧 BTT 被更名为 BTTOLD，新 BTT 按 1:1,000 比例进行置换，总供应量调整为 990,000,000,000,000 枚。该 redenomination 不改变 BTT 的市场价值基础及原始分配比例。

7.2 BTT 代币分布

BTT 的初始代币分布覆盖公开销售、私募、种子轮、空投、团队与基金会、生态建设及合作伙伴等用途。根据原始发行口径，各类别分配比例及按新 BTT 口径折算后的数量如下：

类别	比例	新 BTT 口径数量	用途说明

Public Sale	6.0%	59,400,000,000,000	公开销售
Private Sale	2.0%	19,800,000,000,000	私募销售
Seed Sale	9.0%	89,100,000,000,000	种子轮
TRON Airdrop	10.1%	99,990,000,000,000	TRON 用户空投
BitTorrent Protocol Airdrop	10.0%	99,000,000,000,000	BitTorrent 协议用户空投
Team	19.0%	188,100,000,000,000	团队与 BitTorrent Foundation
TRON Foundation	20.0%	198,000,000,000,000	TRON Foundation
BitTorrent Ecosystem	19.9%	197,010,000,000,000	BitTorrent 生态建设
Partnerships	4.0%	39,600,000,000,000	合作伙伴

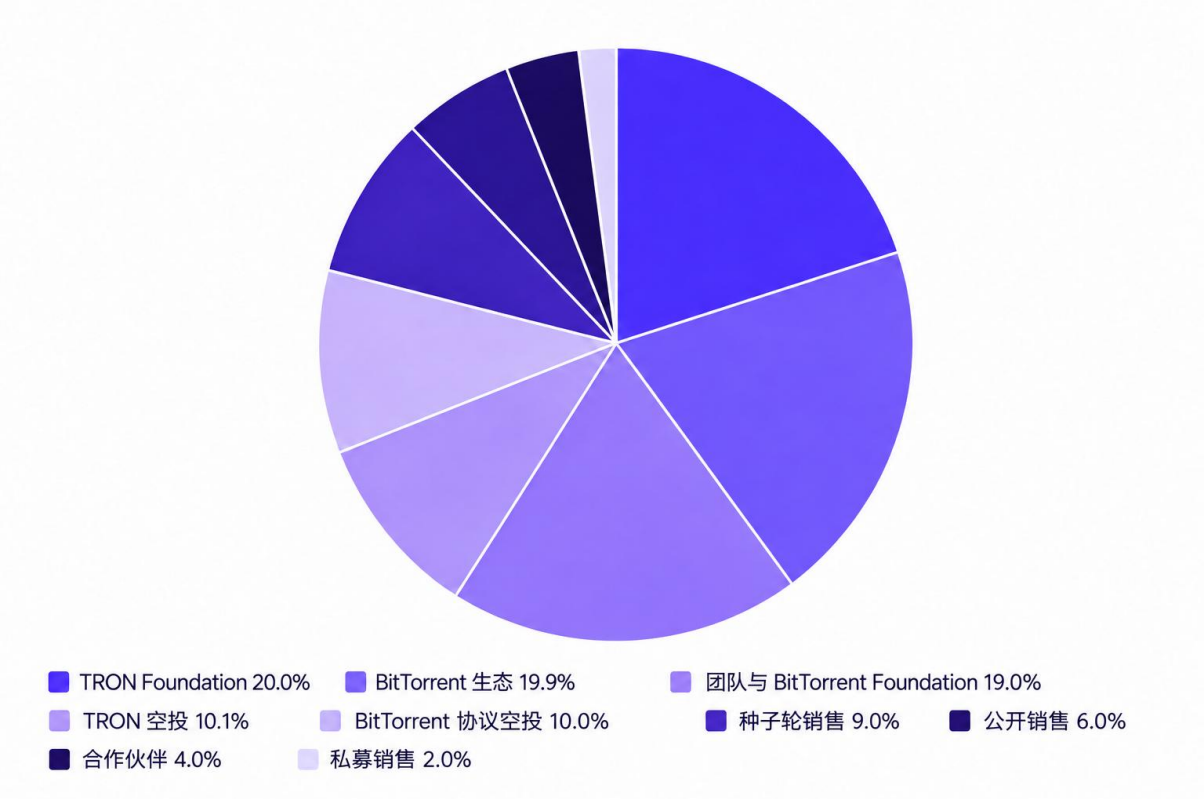


图 7-1 BTT 初始代币分布

7.3 质押与奖励机制

BTTC 网络通过 BTT 质押构建起一套去中心化、可验证的共识与治理体系。参与者可通过三种方式参与网络运行与收益分配：直接质押成为验证人、与他人协作构建验证节点（验证人合伙机制）、或通过委托方式间接参与共识过程。所有质押与奖励分配流程均由链上智能合约自动执行，确保激励机制的公平性与可信度。质押奖励总量设有上限，每年按既定比例逐步释放，并在 2.0 机制下引入了链上动态调整能力，可根据网络运行状况和社区治理决策，灵活优化释放节奏和分配比例。整个激励过程完全由智能合约自动执行，保障公平高效，无需任何中心化干预。

7.3.1 验证人质押机制

验证人是 BTTC 共识层的核心角色，负责在 Delivery 链上参与跨链状态检查点的签名，以及在 BTTC 链上轮值出块。要成为验证人，需在 TRON 主链的 Root 合约中质押至少 1 万亿枚 BTT 并绑定专属 NFT 身份凭证。设计机制如下：

- 权重平等机制：每个验证人“1 节点 1 票”，与质押金额无关，防止投票权过度集中；
- 锁仓与解绑约束：质押 BTT 设有冻结与退出等待周期，防范短期套利与频繁切换；
- 链上透明性：所有质押与激励信息公开可查，便于外部审计与社区监督；
- 经济激励：
    - 签名奖励：在 Delivery 链完成状态签名后，验证人平均分享跨链手续费；
    - 出块奖励：在 BTTC 链轮值出块期间，获得交易手续费收入。
    - 检查点奖励：Delivery 链每轮生成的状态检查点被目标链成功验证后，系统会对该轮参与签名并上传证明的验证人发放额外激励。

所有奖励通过智能合约自动发放，无需人工分配，确保激励分配公开透明、可审计。

7.3.2 验证人合伙机制

为适应实际运营中节点服务能力与代币资本的分离情形，BTTC 原生支持“验证人合伙”机制，允许多个地址协作组成一个验证节点。设计机制如下：

- 多地址联合：多个地址共同质押，系统视为一个验证人节点；
- 奖励自动分润：系统根据各地址质押比例，按签名与出块奖励自动分配收益；
- 链上处理：所有分润流程由合约执行，避免链下博弈；
- 适用场景广泛：适用于 DAO、社区、节点服务商与资本提供方协作共建节点。

验证人合伙机制仅为验证人构成的组织形式扩展，不改变治理结构、不引入新角色，却大幅提升了网络参与灵活性与去中心化水平。

7.3.3 委托质押机制

普通 BTT 持有者无需独立运行节点，也可通过委托质押机制参与网络共识、共享网络激励。设计机制如下：

- 自由委托：持币用户可在 TRON 主链界面自由选择验证人委托；
- 收益分享：验证人需按其获得的签名与区块奖励，按质押占比向委托者分润；
- 锁仓机制：委托操作需遵循链上设定的绑定与解绑等待周期；
- 机制价值：降低用户参与门槛，增强验证人网络稳定性与生态治理活跃度。

委托者虽然不直接参与节点运行，但通过质押行为同样获得链上激励回报，在机制保障下享有公平、透明的经济收益权。

7.4 解锁与释放时间

BTT 已完成 redenomination，以下说明中的历史 BTTOLD 数量可按 1:1,000 比例折算为新 BTT，分配比例及释放逻辑保持不变。

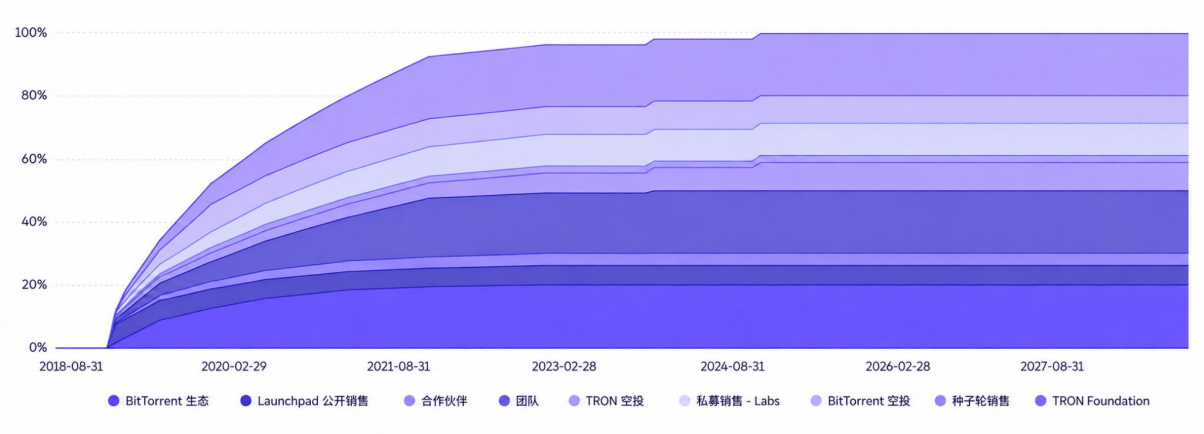


图 7-2 BTT 代币释放进度

类别	总比例	开始释放	完成释放	释放规则
Launchpad Public Sale	6.0%	2019-01-31	2019-01-31	TGE/公开销售完成后一次性释放 6.0%
Private Sale	2.0%	2019-01-31	2019-12-31	12 个月线性释放
Seed Sale	9.0%	2019-01-31	2019-12-31	12 个月线性释放
Team	19.0%	2019-01-31	2021-12-31	36 个月线性释放
TRON Foundation	20.0%	2019-01-31	2021-12-31	36 个月线性释放

TRON Airdrop	10.1%	2019-01-31	2025-01-31	2019 年 1 月 31 日进行首次空投，数量为 2,970,000,000 BTTOLD，占总供应量 0.3%；2019 年 2 月 11 日进行空投，数量为 10,890,000,000 BTTOLD，占总供应量 1.1%；此后自 2019 年 3 月 11 日起至 2025 年 2 月 11 日，按月向 TRX 持有人发放，月度发放量逐年递增，分别为 990,000,000、1,072,500,000、1,155,000,000、1,237,500,000、1,320,000,000、1,402,500,000 BTTOLD。
BitTorrent Airdrop	10.0%	2019-01-31	2029-01-31	前期释放较快，后期小额释放，2029 年 1 月完成 10.0%
BitTorrent Ecosystem	19.9%	2019-01-31	2029-01-31	前期释放较快，后期小额释放，2029 年 1 月完成 19.9%
Partnerships	4.0%	2019-01-31	2029-01-31	前期释放较快，后期小额释放，2029 年 1 月完成 4.0%

7.5 通胀模型

BTTC 的通胀机制设计旨在动态响应网络规模和用户活跃度的变化。初步设定年通胀上限，并依据具体的链上指标（如验证人节点的在线率、网络交易规模、治理投票参与率）动态调整实际通胀率。未来计划引入通证销毁机制，将未被认领的奖励或部分交易手续费定期销毁，以调节代币供给节奏，增强通证稀缺性。通过明确的激励释放规则和治理引导下的经济调节机制，BTTC 确保通胀政策既能满足生态发展需要，又能有效维护通证的长期价值稳定。

7.6 回购与销毁机制

为进一步完善 BTT 的价值捕获与通缩机制，BitTorrent 自 2026 年第三季度起实施长期回购销毁计划。BitTorrent 去中心化业务产生的全部收入将用于按市场价格回购 BTT。

首次回购的 BTT 代币将在 Q4 季度首月中旬转入黑洞地址，进行永久销毁。销毁完成后，BitTorrent 将公布实际销毁数量、占 BTT 总供应量的比例及对应的链上交易哈希，以确保执行过程公开、透明并可独立验证。

随着 BTTInferGrid 等去中心化业务的发展，相关业务收入的增长将相应增加后续回购资金规模，形成由生态业务收入驱动的长期代币供应收缩机制。

8. 应用场景与生态构建

BitTorrent 生态通过 BitTorrent、BTTInferGrid 与 BTTC 的协同，为 AI 推理、模型与数据分发、跨链资产流通及多链治理等场景提供基础设施支持。

8.1 去中心化 AI 推理与存储服务

开发者或应用可以通过 BTTInferGrid 提交 AI 推理请求，由网络根据任务需求、节点性能和资源状态匹配合适的 GPU 节点。节点完成计算后返回推理结果，并由验证机制对任务执行情况和结果可信度进行评估。

任务状态、服务费用和节点收益可以通过 BTTC 进行记录与结算，形成“任务提交—算力调度—推理执行—结果验证—链上结算”的完整服务流程，适用于 AI 应用调用、大模型推理及高并发计算等场景。

8.2 模型、数据集与内容的 P2P 分发

大模型权重、训练数据集和其他大规模文件通常具有体积大、下载时间长和中心化带宽成本高等特点。BitTorrent 基于成熟的 P2P 协议，将文件分割为多个数据块，并通过分布在不同地区的节点共同提供传输，使用户能够同时从多个节点获取所需内容。

模型提供方、开发者和内容发布者可以通过 BitTorrent 网络发布模型、数据集及其他大规模文件。文件哈希用于校验数据完整性，官方发布渠道或数字签名则可用于确认资源来源。BitTorrent Speed 和做种机制能够进一步提高传输效率与资源可用性，为 AI 模型和数据的大规模分发提供基础支持。

8.3 跨链桥与资产流通

BTTC 基于 Delivery 链构建跨链数据与状态中继机制，为不同区块链网络之间的资产和消息传递提供统一通道。平台支持 ERC-20、TRC-20、BEP-20 等主流资产标准，并通过检查点机制完成跨链事件的验证与广播。

用户在源链发起资产锁定后，Delivery 链监听并验证相关事件，生成检查点并提交至目标链。目标链完成验证后释放或生成对应资产，从而实现跨链转移。开发者也可以基于 BTTC 的跨链能力构建多链 DEX、跨链借贷、流动性聚合及其他多链应用。

8.4 DAO 治理与投票系统

随着 DAO 在不同区块链网络上部署，成员身份、治理凭证和投票结果也逐渐分散在多条链上。BTTC 的状态同步与跨链验证机制可以连接不同网络中的治理数据，为多链 DAO 提供统一的状态传递与执行基础。

治理提案可以在指定网络中发起，成员根据治理规则参与投票，投票结果经验证后同步至相关链或治理合约执行。开发者可将跨链投票、资金释放、协议参数调整等操作接入 BTTC，形成从提案、投票到执行的治理闭环。

9. 路线图与生态发展规划

BitTorrent 生态将围绕 AI 与大模型应用，推动模型分发、去中心化推理和链上结算协同发展，逐步构建连接模型开发者、算力提供方与应用团队的开放服务网络。

9.1 阶段路线图

第一阶段：基础服务建设

完善开源模型分发、版本校验和 GPU 节点接入，贯通 BTTInferGrid 推理调用、结果验证与 BTT 费用结算，建立面向大模型应用的基础服务能力。

第二阶段：应用能力扩展

逐步扩展文本、图像和语音模型服务，支持知识库问答、检索增强生成等场景，完善弹性调度与数据访问管理，满足开发者和企业的应用部署需求。

第三阶段：开放生态协同

探索多模型调用、工具协作和连续任务执行，依托 BTTC 完善跨服务计量与结算，支持 AI 智能体应用发展，推动模型、算力与应用服务之间的开放协作。

各阶段将根据技术验证、服务质量和实际需求逐步推进，具体进展随产品发布持续更新。

9.2 生态发展重点

- **模型与算力合作**：连接开源模型社区和 GPU 资源提供方，推进模型适配、性能评测与节点供给建设。
- **开发者与应用支持**：完善 API、SDK、测试环境和示例项目，通过技术合作与阶段性资助支持 AI 应用落地。
- **可持续激励**：围绕有效调用和资源贡献完善 BTT 支付与激励机制，推动网络增长与真实服务需求相匹配。

9.3 安全与治理

持续推进代码与合约审计，完善模型来源校验、推理验证和数据保护机制。通过公开提案、社区讨论及分阶段部署推进重要规则调整，并以服务稳定性、有效调用和应用留存评估生态发展成效。

10. 结语

随着 Web3 与人工智能应用持续发展，模型与数据分发、算力供给、跨链通信和价值结算正在成为新一代数字基础设施的重要组成部分。BitTorrent 生态在既有 P2P 网络与跨链技术的基础上，进一步拓展去中心化 AI 算力能力，致力于连接用户、开发者、节点资源与区块链网络。

在这一架构中，BitTorrent 负责模型、数据集和大规模内容的 P2P 分发；BTTInferGrid 连接全球分布的 GPU 节点，为开发者和应用提供去中心化 AI 推理算力；BTTC 通过三层架构、PoS 共识

和跨链中继机制，为资产、数据和服务状态提供跨链通信与链上结算支持；BTT 则作为贯穿生态的经济层，与各产品线的业务和链上机制相衔接。

三大产品线通过“模型与数据分发—去中心化推理—跨链交互与链上结算”的路径形成协同，使分散的带宽、算力和区块链资源能够服务于 Web3 与 AI 应用。未来，BitTorrent 生态将继续围绕开放接入、安全验证、低成本运行和可持续协作推进技术与产品建设，并根据实际进展逐步完善网络能力和开发者支持体系。

我们欢迎全球用户、算力节点、验证人、开发团队、AI 应用开发者及生态合作伙伴参与 BitTorrent 生态建设，共同探索更加开放、高效、可信的去中心化网络基础设施。