YOM

去中心化 GPU 边缘基础设施

每位用户身边的 GPU，按会话或按小时付费

技术文档与白皮书：网络、协议，以及运营者如何赚取收益。

V1.0 · 2026年11月21日 · YOM.NET/DOCS

目录

问题

后云时代的终局

起源与愿景

解决方案

工作原理

区块链

网络

运营者

收益

赚取

NANO

节点许可证

NaaS 委托

收益最大化

推荐

构建

概览

快速开始

JS 集成

- 自定义身份验证
- 事件处理
- 动态布局
- 简易界面钩子
- 多路串流设置

关于

代币分配

路线图

社交媒体

常见问题

Compute Pass 积分说明

分区

问题

后云时代的终局

说我们需要*更多*数据中心的论调是一个谎言。

世界并不缺算力。它是被错配的算力淹没了。运行实时 AI 和下一代 AAA 游戏所需的原始算力已经存在，其中大部分正关着机，躺在人们的家里。

YOM 不是在建一个更好的数据中心。我们是在构建一个让数据中心变成可选项的层。

沉睡的巨人：家中的闲置容量

世界上大部分消费级 GPU 的算力，在一天中的大部分时间都处于闲置。

数以百万计的 RTX 3090、4080 和高端游戏设备，正在运行浏览器标签页，或者什么也没运行。这是地球上最大的算力池之一，而中心化云服务商把它当作不存在。我们把它当作供给端。

激活这些硬件并不会*增加*容量。它让一种已经制造出来、已经分布各处、已经付过钱的资源投入工作。

而且供给端不只有消费级 GPU。电信铁塔站点上的 GPU，在关键之处增加了电信级的边缘覆盖。YOM 已经在 Cornerstone 的铁塔站点上，与 StonesThro 和 Zadara 一起，在 20 天内运行了 168 场实时游戏会话。Cornerstone 是英国最大的移动通信塔公司（16,500+ 个站点），起源于 Vodafone 和 O2 的网络共享合资企业。这是混合拓扑，而不是单一来源的市场。

中心化云无法跨越的经济底线

要扩张，超大规模云服务商就必须花钱。把容量翻一番，意味着往芯片、混凝土、土地和散热里投入数十亿，这笔抵押般的负担为它们的定价设定了一个硬性的底线。在为高峰预留的数据中心 GPU 上用一小时，费用约为 US$1.23，因为总得有人为那栋建筑，以及它闲置的时间买单。

YOM 没有建筑。

我们不买硬件，不浇筑混凝土。我们在已经存在的硬件之上编排一套协议，这让一场最长一小时的 Standard 会话只需 US$0.08。这不是折扣。这是一种不同的成本结构，对于背负着数据中心资本开支的人来说，在数学上是遥不可及的。

战胜光速

中心化云还在和物理学作战，而且正在落败。

你无法优化光速。如果服务器在一千公里之外，延迟就是内置的，再多的光纤也改变不了这个事实：那台机器离玩家就是太远。YOM 通过无视地理来解决延迟：网络上的每台游戏 PC 都是潜在的边缘节点，这把算力移到了用户自己的街区。信号沿着街道传播，而不是穿越整个国家，网络的目标是在节点位于玩家约 50 km 以内时，延迟中位数低于 10 ms。

这就是通往本地游戏主机响应速度的路径：靠的是邻近，而不是蛮力，实测延迟在 source of truth 上实时公布。

当其他一切都归零时，什么依然稀缺

更深层的转变是，软件本身的价值正在崩塌。任何拥有前沿模型的人，都可以在一个周末里生成自己的应用。当一切都可以被生成，智能就变得充裕，软件也变得廉价。依然稀缺的，是你无法用提示词凭空变出来的东西：**延迟、地域性、能源、硬件和物理存在。**

这就是 YOM 所立足的地基。当生成变得廉价，拥有信息就没那么重要了；在靠近玩家的地方运营低延迟系统，则更加重要。随之而来的经济，奖励的是真实地点的真实贡献者，而不是遥远的垄断者。矿工、零售运营者和电信运营商在同一个底层上贡献。这个网络向任何拥有合格机器的人开放。

这一转变是结构性的

我们正在目睹从寻租式平台向参与式网络的转变。

- **旧世界：** 你付钱给一家公司，租用远方的一台电脑。公司拿走利润；地球为新的制造买单。
- **新世界：** 你付钱给一位对等方，使用附近的一台电脑。贡献者拿走回报；不需要新建任何东西。

YOM 是这一转变的基础设施。我们从云游戏开始，因为游戏是最难的实时串流问题：超低延迟、AAA 画质，以及每位玩家同时涌入的晚间高峰。解决了游戏，同一个网络就能在同样的节点、同样的会话上，服务实时 AI 对话，语音优先。

第一步是游戏。第二步是实时 AI。

起源与愿景

YOM 并不是诞生在会议室里。它是在生产环境中锻造出来的。我们的创始人 Jorrit Velzeboer 经营过一家 3D 串流机构，在那里他遇到了在中心化云上串流 GPU 工作的成本：尽管做了极致的优化，每个用户小时都要承担一块数据中心 GPU 的价格：这块 GPU 远离用户，并且在等待高峰期间也要为它付费。中心化云是为企业工作负载打造的，不是为大众市场的交互式串流打造的。所以我们着手解决一个超大规模云服务商没有理由去修的问题。

时间线

- **2021：** 构建了自研的像素流后端。这个 MVP 表明，靠近玩家的消费级 GPU 可以串流游戏，整家公司都建立在这个洞见之上。
- **2022：** 得出结论：只有一个由运行它的人所拥有的 GPU 网络，才能以具有成本效益的方式扩展。Rohan Solanki 加入，负责设计 HyperOrch（YOM 的边缘编排器）和 NodeOS（节点操作系统）的架构。
- **2023：** 荣获欧盟委员会的卓越印章（Seal of Excellence），表彰其深度科技创新。与 Andrew Pringle 和 Jeff Outlaw 展开了首批上市推广的对话。
- **2024：** 与游戏行业的资深人士一起，正式确立了领导团队。HyperOrch alpha 串流了它的第一个 UE5 演示。与 Outlier Ventures 和 Borderless Capital 完成 Pre-Seed 轮融资。
- **2025：** 发布了第一批 NANO。获得 Gamescom 2025 最佳科技初创企业奖。
- **2026：** 6 月代币和主网上线。Walter 串流玩家已经拥有的 Steam 和 Epic 游戏。46,000+ 个注册账户。在 Cornerstone 的铁塔站点上，YOM 在 20 天内运行了 168 场实时游戏会话。生产网络在 source of truth 上实时公布距节点 50 km 以内玩家的延迟中位数；目标是 10 ms 以内。

一个由运行它的人所拥有的网络

我们正在朝着一种实时参与的经济构建，而不是抽象凭证或中心化智能的经济。

随着互联网从**文件的互联网**演进为**事件的互联网**，价值也随之转移。智能变得充裕；软件变得廉价。依然稀缺的，是抽象无法抹去的东西：**延迟、地域性、能源、硬件和物理存在。** 在那样的世界里，权力不再属于拥有信息的人。它属于能够在物理和感知的硬性限制之内，在接近人类时间的尺度上运营实时系统的人。

YOM 为这样一个世界而存在：

- **人们是运行数字体验的基础设施的运营者和所有者**，而不仅仅是它的用户。
- **价值流向贡献者，而不是平台。** 我们用一种直接的经济，取代对遥远垄断者的依赖，它奖励人们所提供的硬件、能源和用心。
- **贡献者基础是开放的。** 矿工、零售运营者和电信运营商在同一个底层上运行节点。这个网络向任何拥有合格机器的人开放。
- **数字现实保持扎根：** 一个必须由真实地点的真实贡献者来运行、测量、验证和付费的系统，而不是一场供人消费的奇观。

这不是理论。一台宿主 PC 通过向附近的玩家串流算力来赚取收益。在网络的 50% 利用率目标下，一路 Standard 串流每月赚取约 ~US$53；后端估算器根据每个宿主的实测硬件计算它所支持的容量（实时数字见 /operators）。

使命

把本地拥有的现实世界硬件，变成活跃的经济基础设施。

我们协调一个由社区所有的边缘设备组成的全球网络，由我们的边缘编排器 HyperOrch 来编排，并通过与运营者的收入分成来给予回报，让它在成长的过程中变得更加分布式。这是中心化云的反面。

我们从云游戏开始，因为它是最难的实时串流问题：超低延迟、AAA 画质，以及每位玩家同时涌入的晚间高峰。像素流天然生长在边缘，地域性就是整场游戏的全部。解决了这个，同一个网络就能服务实时 AI 对话，语音优先，以及中心化数据中心触到天花板的其他所有实时工作负载。

我们存在的目的是：

1. 在自动化认知的世界里，让**人的能动性在经济上保持价值**。
2. 把**闲置的硬件转化为有报酬、可验证的生产容量。**
3. 把数字经济锚定在**地域性和在线时间**上，而不是遥远的资本集中。

领导团队

我们把深度科技工程与 AAA 游戏商业经验结合在一起。

- **Jorrit Velzeboer（创始人兼 CEO）：** 连续创业者，前风险投资人。设计了 YOM 的商业模式和技术方向。
- **Jeff Outlaw（联合创始人兼 COO）：** 曾任 Oculus VR 产品经理。游戏行业资深人士，把 YOM 与工作室和玩家社区连接起来。
- **Rohan Solanki（联合创始人兼 CTO）：** HyperOrch 和 NodeOS 的架构师。大规模分布式系统专家。
- **Andrew Pringle（联合创始人兼 CCO）：** 数十年的零售、商业和分销战略经验。
- **John Kavanagh（业务拓展总监）：** 20+ 年的 IP 和娱乐行业人脉。负责发行商和批发需求。
- **Jeroen Elout（CXO）：** 增长和用户体验专家；负责 Walter 的转化和留存。

顾问包括 OnLive 前 COO 及 Eidos Interactive CEO、前 PlayStation Store 全球负责人，以及 PLAYERUNKNOWN Productions 的 CEO。

我们在构建未来的基础设施，立足于人的价值。

分区

解决方案

工作原理

运行一个面向交互式工作负载的去中心化边缘网络，需要三件事协同工作：一套过硬的技术栈、链上经济，以及让玩家和开发者都觉得毫不费力的体验。

YOM 构建了自己的编排、节点操作系统和串流栈，让串流安全、高效且易用；这项技术正在申请专利。下一章介绍架构。

当你点击“Play”时

从玩家的角度看，没有下载，也没有安装。你点击一个链接，几秒钟内就能开始玩，游戏在远程渲染，并串流到你的浏览器或设备上。对玩家来说，这意味着：

- 即点即玩，即使是低配硬件。
- 在 Walter 上玩你已经拥有的游戏，每月 US$4.99 起。
- 靠就近部署实现的低延迟：距节点 50 km 以内的玩家，延迟中位数目标低于 10 ms，在 source of truth 上实时测量。

对工作室和发行商来说，去掉安装、补丁和硬件限制，能打开广阔得多的受众和更高的转化率。

实际发生了什么

在技术层面，当你点下“Play”时发生的是：

1. 玩家通过网页应用、嵌入链接或合作伙伴门户请求一款游戏。
2. 我们的边缘编排器 HyperOrch，把会话发送到最近的、具备所需 GPU 能力且有空闲容量的节点。
3. 那个节点，也就是由 NANO 启动的游戏 PC，或运行同一套节点软件的电信机架，在一个全新的容器中启动游戏，用 GPU 的硬件编码器（NVENC）对其编码，并通过 WebRTC 串流，所以玩家无需安装任何东西。
4. 游戏画面从玩家附近的节点以点对点方式串流，网络在有节点的地方，距离在约 50 km 以内。
5. 会话结束时，收入自动分配：
    - 60% 归节点运营者
    - 30% 用于网络的开发和增长
    - 5% 分给所有继续持有自己收益的人
    - 5% 注销，减少代币供应量
6. 每场会话都通过智能合约结算。

YOM 基金会运行结算，并治理扩大供应量的计划，再一步步把控制权移交给社区：先是代币持有者对供应计划的投票，然后是选举产生的理事会成员。

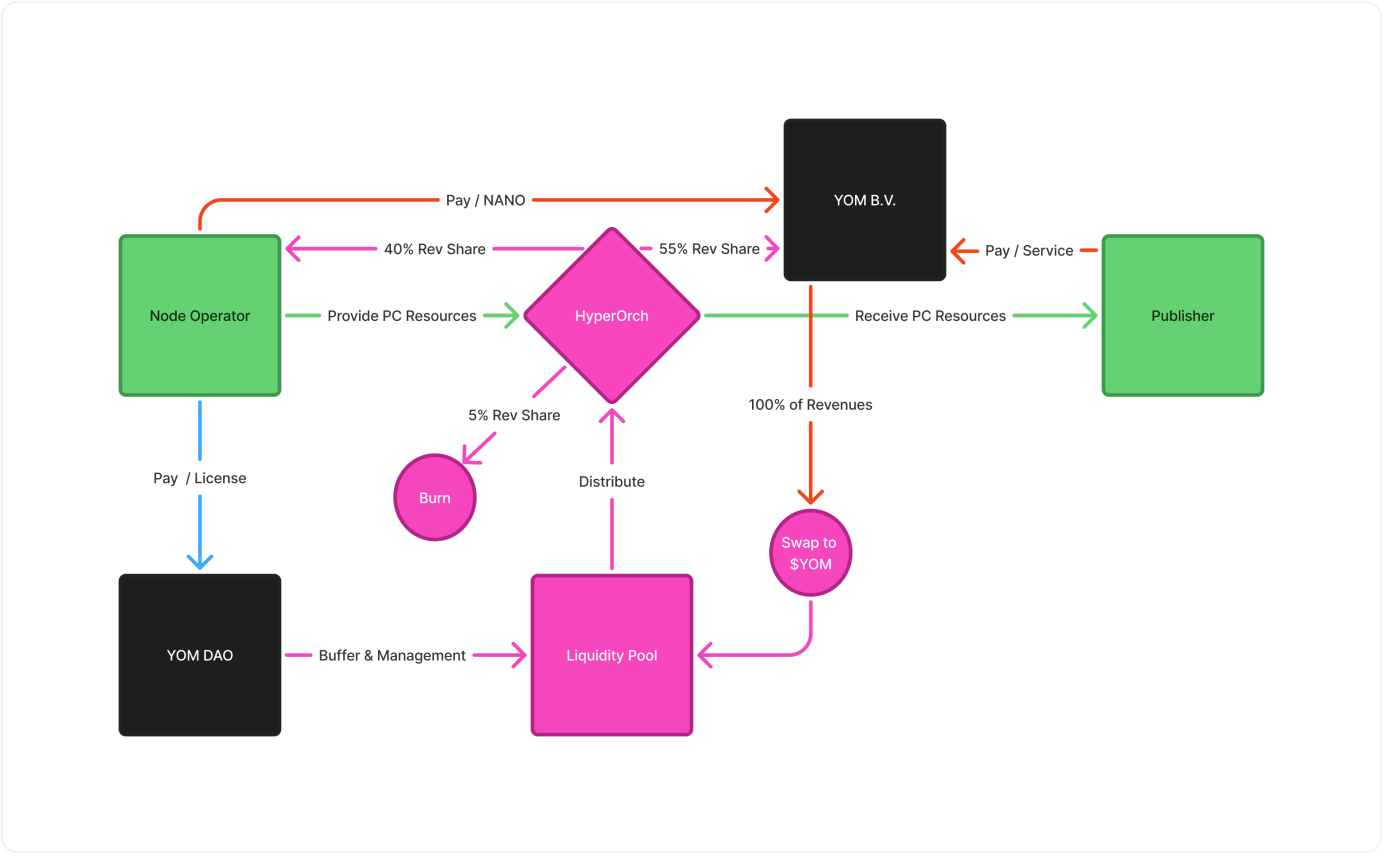
一个自我调节的经济

让 YOM 保持平衡的，是一个循环的、自我调节的结构：玩家创造需求，节点满足需求，发行商为其出资，每一位参与者都分享成果。

价值（$YOM）按固定的每会话规格价格和协议层面的分配流动：运营者奖励、网络开发、社区份额和注销。每笔会话费用都自动结算。

治理分阶段移交给社区。代币持有者的投票将按 $YOM 持仓和 XP 加权，XP 是一种不可转让的贡献分数，通过为网络做贡献获得。许可证随着需求填满网络而分批发放，所以新增供给跟随真实的使用量。结果是：

- 运营者按所服务的会话获得报酬，自动结算，按固定的规格价格。
- 未通过验证的节点会被自动隔离。



面向节点运营者

运营者使用 NANO 贡献自己的硬件，NANO 是一个即插即用的启动设备，几分钟内就能把一台普通游戏 PC 变成网络节点：它从自己的加密驱动器启动一个只读系统，与所有者的操作系统相互独立。

运营者的收益来自：

- 他们所服务的每笔会话费用的 60% 份额，以美元计价，以 $YOM 支付：在网络的 50% 利用率目标下，每台 NANO（一次一场 Standard 会话）每月约 ~US$53，未扣除电费。总收益取决于后端估算器计算出的宿主容量（实时数字见 /operators；明细见收益最大化），扣除他们自己的电费。
- 加入一个由社区所有的经济，他们控制自己的硬件和自己的时间。
- 为可靠的机器和社区工作赚取 XP，这是一种不可转让的贡献分数，将为代币持有者的投票加权（见 XP）。

运营者可以随时自由加入或离开，这使供给自然地与需求保持一致。

面向发行商与工作室

YOM 为发行商提供了一条分发和变现交互式内容的新渠道。去掉基础设施和硬件的门槛，能扩大受众，并降低分发成本。

发行商可以：

- 带来他们的构建（任何引擎制作），把可玩的会话放进他们自己的网站、应用和渠道：一个嵌入链接、一个带他们自有品牌的页面，或者作为托管会话运行的他们自己的串流器（构建概览）。
- 把基础设施成本，从为高峰预留的数据中心 GPU 上一小时约 US$1.23，降到按会话计的算力规格，每场从 US$0.04 到 US$0.24，一场会话最长一小时。
- 完全掌控变现方式：买断、订阅、广告支持或免费游玩。

有了 YOM，发行商触达新的受众，而无需背负数据中心的成本。

区块链

当玩家按下 **Play** 时，一个节点被唤醒，一块 GPU 转动，数据包流动，价值随之转移，并全程在链上结算。这一切的核心是 **$YOM 代币**：算力的记账单位，运营者的报酬，以及随每场会话而不断销毁的供应量。

YOM 今天在 **Avalanche C-Chain** 上结算，构建为 LayerZero 全链代币（OFT）；**Base** 部署是下一步。全网使用单一的全球 $YOM 代币。这是一个有意的选择：以目前的交易量，Avalanche C-Chain 绰绰有余，过早迁移只会让流动性碎片化。带区域代币的主权 L1 在路线图上（见下文），并不是当前的状态。

算力协调，而不是通用公链

$YOM 不是一个通用智能合约的押注。它是一个全球 GPU 网络的结算与协调层。每一个渲染的帧、每一场串流的会话，都是由算力、带宽和奖励构成的微型经济。

在底层，YOM 的边缘编排器 **HyperOrch** 会把每场会话发送到最近的、具备所需 GPU 能力且有空闲容量的节点。节点在它自己的容器里启动游戏，会话串流给玩家，费用在链上原生结算。

奖励与**所服务的会话**挂钩，而不是与验证的区块挂钩，并通过固定的算力规格以美元计价（见收益）。这让网络能够支付稳定、可预测的奖励，而不依赖代币的现货价格。运营者与协议签约；协议把 60% 分给运营者，30% 分给网络开发，5% 分给所有继续持有自己收益的人，并注销 5%。

运营者通过服务会话来赚取收益。另外，**每笔会话费用的 5% 分给所有继续持有自己收益的人**。在最初的交易期内，生态池为去中心化交易所上的流动性奖励提供资金。

天然通缩

这个代币有严格的通缩结构：**永远不能再铸造新代币**（供应量固定）。每场会话，费用的 **5%** 被注销，永久减少供应量。随着使用量增长，流通量变得更稀缺，使代币持有者的利益与网络活动的持续保持一致，而不是与投机挂钩。

管理与治理

协议和代币由 **YOM 基金会**管理，它负责运行网络的结算，并治理扩大供应量的计划。基金会运营国库，并聘请专业做市商为代币提供流动性。

控制权分阶段移交给社区。首先，代币持有者对供应计划投票：运营者激励费率、国库分配、硬件标准和资助。然后，社区选举产生的理事会成员，从上线后的第二年起，开始接替今天的创始人代表。投票的权重将 **67% 由 XP 决定**（XP 是一种不可转让的贡献分数），**33% 由 $YOM 持仓决定**，这样引导网络的，是做事的人，而不只是最大的持有者。

审计与合约管理

独立的智能合约安全公司 **Hashlock** 于 2026 年 4 月审计了代币合约代码，于 2026 年 10 月审计了收益和质押合约代码。每份报告都是公开的，写明所审查的代码版本，并列出每一项发现及其状态：Hashlock 关于 YOM 的报告。

代币合约的管理权限位于 **2-of-3 多重签名**和 **48 小时时间锁**之后，所以一项变更在生效之前，就能在 Avalanche 上看到。任何人都可以在公共浏览器上核查这两者：时间锁和多重签名。代币合约是 `0xb6314518b61b4864162c7aE7fdc36261e0A14C4b`。

路线图：主权 L1 与区域代币

随着交易量扩大，YOM 计划迁移到一条**主权 L1**，并引入**区域代币**（例如 $yEU、$yCHN），每个区域代币与 $YOM 之间是销毁/铸造关系，使本地市场能够独立为算力定价，同时 $YOM 在全球范围内捕获价值。这一迁移的前提是：交易量达到这样一个点，即共享链上的 gas 超过了能证明一条专用 L1 合理的阈值，目标是 **2027+**。在那之前，所有结算仍然留在 Avalanche C-Chain 上，使用单一的全球 $YOM。

关于完整的代币模型、分配和 TGE 的详细信息，见代币分配。

网络

YOM 是一个去中心化的 GPU 边缘网络，旨在把全世界闲置的消费级 GPU 编排成一张算力网格，让每场会话由靠近用户的节点提供服务（在网络有节点的地方，距离在约 50 km 以内），成本远低于中心化云。游戏是第一个工作负载，因为它最难：毫秒级的延迟预算、AAA 画质，以及无法暂停或重试的有状态会话。同一个网络把实时 AI 对话作为会话来运行，合作伙伴自己的串流器或模型一经审核并固定版本，也走同一条会话路径运行。

这套技术栈涵盖自研的像素流方案、安全的裸机边缘计算，以及 YOM 的边缘编排器 HyperOrch。这些组件合在一起，把游戏串流到任何带浏览器的设备，边缘延迟低，从启动到串流全程安全，成本只有中心化云游戏的一小部分。

该协议由三个组件构成：**HyperOrch**（边缘编排器）、**NodeOS**（节点的只读操作系统）和 **Universal Streamer**（基于浏览器的像素流）。节点运行在**裸机上，没有 Kubernetes，也没有虚拟机**，所以每场会话到达 GPU 的路径中没有虚拟机管理程序挡路。

天生有状态

大多数分布式 GPU 网络出售的是用于批处理和无状态任务的 GPU 小时：它们把算力路由到任何空闲的 GPU。实时会话从根本上不同。它们需要有状态的执行：游戏状态跨帧持续存在，输入循环以毫秒计。

YOM 是为会话而构建的。同一个会话模型承载游戏和实时 AI 对话，包括合作伙伴自己的串流器或模型，每一场都为一位用户而保持，并紧挨着这位用户。这捕获了只做无状态任务的网络在结构上无法服务的实时工作。

点对点像素流

YOM 部署自己的轻量级像素流服务，在远程的消费级 GPU 上运行游戏，用 GPU 的硬件编码器（NVENC，H.264）对每一帧编码，并通过 WebRTC 串流回玩家，所以玩家无需安装任何东西。会话一经批准，游戏画面会在可能的情况下，通过节点与玩家设备之间的直接点对点连接传送；当防火墙或 NAT 阻断直连路径时，则通过 TURN 中继传送，这样会话仍然可以连通。

视频、音频和输入在传输中使用 WebRTC 标准的 DTLS-SRTP 加密，把跳数降到最低，并让串流在公共网络上保持私密。

Universal Streamer 与中间件

每个 YOM 节点都运行一个**中间件层**，它编排完整的会话生命周期：从准备游戏、启动由 GPU 加速的容器（Docker 配合 NVIDIA Container Toolkit），到管理隧道、健康检查和会话拆除。中间件通过出站连接从 HyperOrch 拉取指令，所以节点不需要开放任何入站端口，并确保每个容器都受到监控，并在几秒钟内准备好串流。

在每个游戏容器内，由 **Universal Streamer** 接手。这个轻量级的串流客户端处理玩家与游戏交互所需的一切：WebRTC 视频和音频、低延迟输入捕获（鼠标、键盘、手柄、麦克风）和自适应码率串流，全部直接在浏览器中运行，无需安装任何东西。

Universal Streamer 让 YOM 与游戏无关。玩家带来他们已经拥有的 Steam 和 Epic 游戏库；工作室上传一个构建。无需集成 SDK。串流器包裹游戏渲染出的画面，与玩家的浏览器建立对等连接，并开始串流。对开发者来说，接入一款新游戏，简单到只需上传一个构建并嵌入一个 iframe。

中间件和 Universal Streamer 合在一起，构成了每个节点的运营骨干，把原始的 GPU 算力大规模转化为基于浏览器的会话。

节点验证与遥测

每个节点都向网络注册，并报告它的 GPU 级别、位置和负载。NANO 节点在每次启动时都会对照已知哈希检查自己的启动文件；HyperOrch 只考虑通过检查的节点，未通过的节点会被自动隔离。

在实时运行期间，节点报告 FPS、延迟、码率、丢包和温度，而免费的 rig check 会在一台机器服务任何会话之前，就测出它能承载多少场会话。

每场会话都在一个干净的容器中运行，会话结束时该容器即被丢弃，所以一位用户的任何东西都不会带到下一位用户。NANO 的磁盘是全盘加密的，每次启动时只能通过 YOM 的密钥服务解锁；被 YOM 撤销的设备会保持锁定。

NANO 节点从它们自己的驱动器启动一个最小化的只读 Linux 镜像，与所有者的操作系统相互独立。首次启动时，该操作系统会：

- 自动检测硬件并安装经过签名的 GPU 驱动，让独立显卡以全性能暴露出来（无需用户干预）。
- 运行一个注册模块，把机器关联到它的运营者，并向 HyperOrch 报告它的 GPU 能力、位置和容量。
- 让每场会话在它自己的容器中运行，会话结束时容器即被销毁，并通过驱动程序的时间切片与其他会话共享 GPU。

HyperOrch：边缘编排器

YOM 网络上的工作负载分配，由 YOM 的边缘编排器 HyperOrch 处理，它今天已在生产环境中运行。

HyperOrch 把每场会话发送到最近的、具备所需 GPU 能力且有空闲容量的节点。HyperOrch 会筛选出在线的、拥有该会话的游戏、具备合格 GPU 且有空闲容量的节点，然后把启动指令发送给其中最近的一个。没有合格 GPU 或没有空闲容量的节点会被跳过，所以最近的节点可能输给一个符合条件的节点。

在节点上，今天有多场会话通过 GPU 驱动程序的时间切片共享同一块消费级显卡。

如果被选中的节点没有响应，下一个符合条件的节点会接手这场会话。如果某个节点在会话中途掉线，玩家会在下一个附近的节点上重新开始。

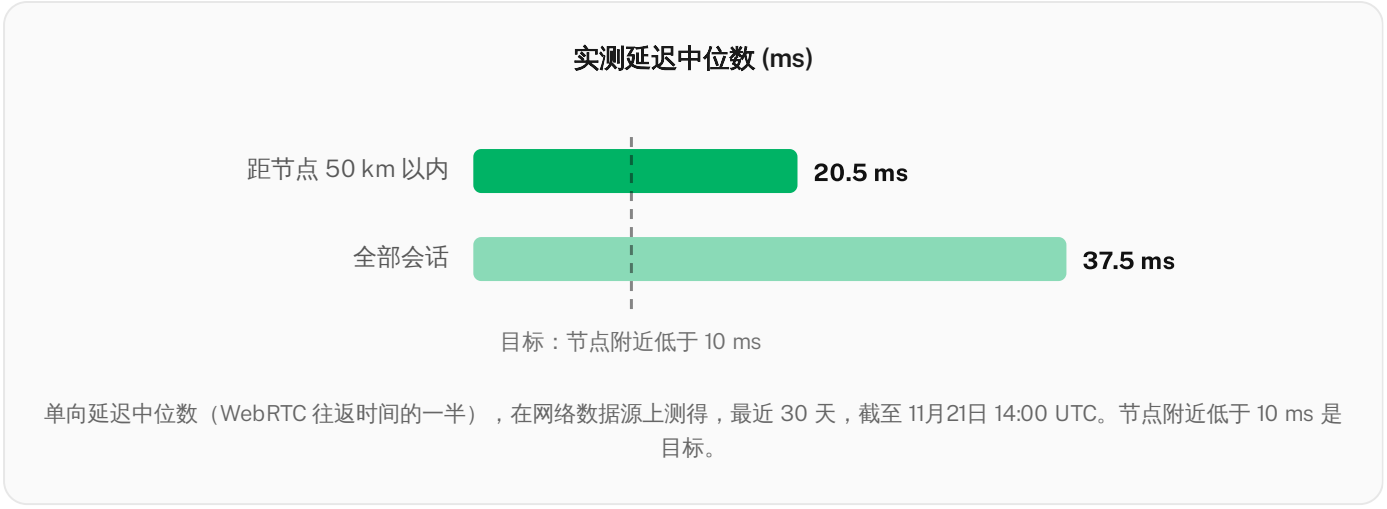
溢出容量与服务等级

网络的主要供给是由社区运营的消费级 GPU，但在任何新区域，密度都需要时间来建立。在运营者供给薄弱的地方，Walter 会话运行在按需租用的 GPU 上：计划第一年约占 Walter 会话的 13%，随着运营者覆盖的扩大，到第五年降至 3%。

服务等级决定每场会话背后的供给，每一级都表述为会话完成率：一小时的会话中，没有因网络原因而断线、顺利结束的比例，而不是正常运行时间。在家用 GPU 上（Basic，尽力而为），设计目标是 98% 的会话没有因网络原因而掉线。Protected 的目标是 99.5% 的一小时会话没有因网络原因而掉线，其会话大部分在电信机架上，而 Carrier 的目标是仅在电信机架上达到 99.9%。

专利

涉及安全启动介质、像素流和编排的技术正在申请专利。



单向延迟中位数（WebRTC 往返时间的一半），在网络数据源上测得，最近 30 天，截至 11月21日 14:00 UTC。节点附近低于 10 ms 是目标。

运营者

在 YOM 生态中，**节点**是去中心化 GPU 边缘网络的骨干。每个节点都是运营者所拥有的一块 GPU，用来串流实时游戏会话。与那些常常闲置、或依赖投机性代币增发的传统加密“节点”不同，我们的节点执行真正的工作，并按所服务的会话获得报酬。

什么是节点？

节点是为网络提供服务的运营者的机器：一台游戏电脑或一台服务器，它的 GPU 为身边的玩家运行会话。当玩家在网络上启动一款游戏时，这款游戏在节点运营者的硬件上渲染，并实时串流到玩家的设备上，每场会话都在它自己的容器中，会话结束时容器即被销毁。

简单来说，运行一个 YOM 节点，有点像托管一个玩家连接上来玩一场会话的小型游戏服务器，只不过玩家只能看到视频画面。由于 YOM 是去中心化的，这些节点可以由普通运营者在自己的硬件上运行。你的节点每服务一场会话，你就为这场会话获得报酬。

任何拥有合适硬件和节点许可证的人都可以运行节点。一台机器的容量由它的 rig check 测得，以单位计：一场 Light 会话占用一个，Standard 两个，Premium 四个，Ultra 六个。一个节点许可证解锁一个单位的容量，每个 NANO Node Runner 都包含两个许可证（足够承接一场 Standard 会话），额外的许可证让一台机器更多的容量投入服务，上限是它的 rig check 所显示的容量。

许可证是网络控制供给的方式。当按提前三个月预测的利用率达到 60% 时，新的 Node Runner 批次和节点许可证就会开放，这样新增容量跟随需求，并让已经在网络上的宿主保持忙碌。

每个宿主能承接多少场会话？

一台物理机器可以**同时服务多场会话**，只要组合装得下它的容量，并且它的资源和带宽能够支撑。从一台 NANO Node Runner 开始，并在增加节点许可证之前，用免费的 PC 检测评估容量。

每场会话都在它自己的容器中运行。拥有足够容量的运营者，可以从一台机器上同时服务多位玩家。

运营者收益

运营者获得每笔会话费用中的固定份额，以美元计价，以 $YOM 支付。在网络的 50% 利用率目标下，一台 NANO（一次一场 Standard 会话）赚取**每月约 ~US$53**，未扣除电费。宿主的总收益取决于后端估算器为实测硬件分配的容量。一台 NANO Node Runner 售价 **US$349**，包含两个许可证；一个新的节点许可证售价 **US$125**。在网络的 50% 利用率目标下，回本时间约为 **~7 months**。

在网络的 50% 利用率目标下	每月收益	硬件	回本
一台 NANO，一场 Standard 会话	~US$53	US$349	~7 months
实测的宿主容量	后端估算器的结果	US$349 + 每增加一个许可证 US$125	见 /operators

收益随你的机器所服务的会话而增长。节点许可证留在你的账户里，所以你可以把它们转移到另一台机器上。

XP

XP 是一种不可转让的贡献分数，将为代币持有者的投票加权。它认可可靠的机器和社区工作；它不会改变会话分配或收益。XP 通过以下方式获得：

1. **网络运营。** 运行稳定、高在线率机器的节点所有者，按其节点的可靠性相应积累 XP。
2. **社区参与。** 积极参与 Discord 或论坛，撰写高质量的指南或教程，以及在社交媒体上宣传，都会产生 XP。Web 应用和 Discord 让成员可以方便地参与能赚取 XP 的活动。
3. **社区任务。** YOM 基金会或社区委员会举办的定期活动，可以为压力测试新功能或帮助新成员入门等任务提供额外的 XP。

收益

YOM 上的节点每服务一场游玩会话，就获得一笔费用，价格由该会话的工作负载规格固定。没有闲置代币挖矿，也没有按分钟计量：会话是不可再分的单位，玩家消费它，运营者服务它，协议结算它。同一个量贯穿整个系统，这就使每一笔收益都可以端到端审计。

算力规格：交付的体验，而不是硬件参数

每场会话都按工作负载规格设定的固定每会话费用结算。规格是一份**交付体验的契约**：玩家实际得到的分辨率、帧率和响应速度，而不是一个 GPU 型号名称。网络在分配任何规格之前，会先测量一台机器能否履行该规格的契约。

规格	每场会话的 Walter 积分	每场会话的网络费用
Light	1	US$0.04
Standard	2	US$0.08
Premium	4	US$0.16
Ultra	6	US$0.24

每个规格的分辨率、帧率和资源分配，见当前的算力规格。Walter 积分支付的是消费者的访问；网络费用支付的是算力会话。订阅价格和网络费用是两笔独立的金额。

费用随会话所预留的资源而变化。在会话时长和利用率相同的情况下，等量的预留容量在各个规格之间获得相同的每会话费用。每款游戏都会被分配一个规格，运营者不能选择；目前每款游戏都按 Standard 规格运行。宿主的实际收益还取决于需求、会话时长，以及它的硬件所能服务的组合。

每场会话，无论是游戏还是实时 AI 对话，都按会话计费，一场会话最长一小时：一场 Standard 会话费用为 US$0.08，而同样一小时在为高峰预留的数据中心 GPU 上约为 US$1.23。

会话积分与游玩时长

Walter 的 Compute Pass 包含每月积分额度。一场会话消耗上面显示的、对应其规格的积分，最长可持续 **60 minutes**。US$9.99 的订阅包含 70 积分，玩家可以把它们用在不同的规格上。额外积分每个 US$0.175。

会话上限描述的是单场会话最长可以运行多久。预测使用假设的时长 20 minutes，来估算典型的使用量和每小时收入；它并不决定玩家的会话时长。在上限之内，更长的会话会把同一笔会话费用摊到更多的游玩时间上。

价值流向

每场通过 YOM 协议结算的会话，都支付该规格的费用，网络在会话关闭时把它分配出去。运营者与**协议**签约，而不是与单个发行商签约，这让无论是谁卖出了这场会话，收益都保持稳定且可预测。

利益相关方	分配	说明
节点运营者	60%	直接支付给服务该会话的运营者。
网络开发	30%	为网络的开发和增长提供资金。
持有者份额	5%	分给所有继续持有自己收益的人。
注销	5%	从供应量中永久注销。

这个 60% / 30% / 5% / 5% 的分配是协议的经济不变量。它统一适用：Walter 会话、企业会话，以及任何通过网络结算的第三方聚合方，都按同样的规格、支付同样的分配。

两部分的收益

运营者的收入有两个组成部分，当你在决定是否插上设备时，这个区别很重要：

1. **使用量收益**：你的机器所服务的每笔会话费用中的 60% 份额。它随网络需求和你的机器的实测容量而增长。
2. **每月奖金**：每台 NANO 每月 US$74，在会话收益之外另行发放。机器每天在线至少 6 小时时，其中 US$37 有保障，另外 US$37 进入一个共享池，按每台机器所服务的会话来分配。它会一直保留，直到更好的补贴机制取代它，我们会提前公布。

在网络的 50% 利用率目标下，一台 NANO（一次一场 Standard 会话）每月赚取约 ~US$53，未扣除电费。宿主的总收益随后端估算器为实测硬件分配的容量而增长；实时数字见 /operators，并随网络需求而变化。

可预测的重新定价，而不是悄悄漂移

会话费用按假设的会话时长进行校准，然后与观察到的使用情况比较。随着游戏库的改善和玩家停留更久，每会话费用可以被重新校准，使每台机器每小时的运营者收益保持稳定。**费用变更会在生效之前公布**：运营者应当能够围绕自己的收益费率来规划，就像围绕自己的电费账单来规划一样。

定价

会话费用以美元计价，以 **$YOM** 支付，使用单一的全球代币。这让运营者的奖励保持稳定且可预测，而不依赖代币的现货价格。代币的详细信息见代币经济学。

为什么这在结构上低于中心化云

终局不是折扣；而是一条不同的成本曲线。数据中心 GPU 每小时 US$1.23 的预留价格，由五年折旧期内的 GPU 资本开支、数据中心的房地产、散热和企业级网络构成：无论有没有人在玩，这些成本都要承担。YOM 的供给是**已经存在的边际消费级芯片**：网络每小时的成本，是运营者的电费，加上让插上设备值得的激励，而且用的是本来就为游戏而买的硬件。

在已有的消费级硬件上服务会话，降低了每场会话需要收回的资本成本。运营者的收益仍然取决于需求、利用率和实际服务的会话时长。

总之：**你的节点通过服务会话来赚取收益，价格由工作负载规格决定**，采用统一、可预测的分配，不取决于是谁卖出了这场会话，也不取决于玩家在哪里。

分区

赚取

nano

我们的即插即用 **NANO** 是一块可启动的加密 SSD，把游戏 PC 变成网络节点，你拥有完全的控制权，设置也很简单：无需安装复杂的软件就能部署你的节点。NANO 售价 **US$349**，包含 **2 个节点许可证**，足够承接一场 Standard 会话。它由我们的硬件合作伙伴 **SIMMS** 制造和分销，并通过授权经销商销售。

如果你不想使用专用设备，符合条件的设备也支持**纯软件的节点设置**。

步骤 1：rig check

免费的 rig check 会判断你的设备是否符合条件，并测出它的容量：网络可以同时调用它多少算力，以单位计。作为参考，它会查看：

系统要求：

- **GPU：** 桌面版 NVIDIA RTX 显卡。显存越大，容量越大。
- **CPU：** Intel i5/i7 或 AMD Ryzen 7/9，以保证多任务处理和稳定性。
- **内存：** 较小的配置 16GB 就够了；推荐 32GB 或更多。
- **网络：** 有线以太网连接，实测上传速度至少 20 Mbps。
- **散热：** 良好的气流和散热，防止过热。

步骤 2：订购 NANO

rig check 确认你的系统符合条件后，你就可以以 **US$349** 订购 **NANO**（参考价格；可能需要加收各地区的税费）。该设备由我们的硬件合作伙伴 **SIMMS** 制造和分销，并通过授权经销商的网店销售，例如 Freshminers.com。

步骤 3：设置你的节点

设备到货后，按以下步骤让它运行起来：

1. **把 NANO 连接到你的 PC。**
2. **从 NANO 启动。**
   - 重启你的 PC，在启动时按 `F2`、`F12` 或 `DEL` 进入 BIOS。
   - 把 YOM SSD 设为首选启动设备。
   - 保存并退出 BIOS 设置以继续。
3. **按照屏幕上的设置引导操作：**
   - 系统启动进入 YOM Node OS 后，按照引导设置操作。
   - 用运营者应用扫描包装盒上的二维码，把这台机器关联到你的账户。应用会为你创建钱包。

就是这样：你的节点已准备好开始服务。

节点许可证

设置好你的 NANO 之后，节点许可证决定了你机器的容量有多少投入服务。容量是网络可以同时调用一台机器多少算力，以单位计：一场 Light 会话占用 1 个单位，Standard 占用 2 个，Premium 占用 4 个，Ultra 占用 6 个。你的显卡、处理器、内存和上传速度决定你机器的容量，免费的 rig check 可以测出它。一台机器可以承接任何装得下的会话组合。宿主需要有线连接，实测上传速度至少 20 Mbps。

一个许可证解锁一个单位的容量。你的机器最多承接它自身的容量，或者它的许可证所解锁的容量，取两者中较低的那个。许可证限制的是投入服务的容量，而不是玩家的数量：5 个许可证可以同时承接一场 Premium 会话和一场 Light 会话，或者两场 Standard 会话和一场 Light 会话，或者五场 Light 会话。每台机器可使用的许可证数量没有上限，机器越强，容量越大。每台 NANO 都包含 **2 个许可证**，足够承接一场 Standard 会话。

这何时生效，以及它对你已经持有的许可证意味着什么：从插槽到容量：每个许可证变成两个。

许可证是在网络上运行容量的权利，它一直留在你的账户里。

在网页应用的节点管理视图中，你把许可证分配给你的 NANO。

如何获得节点许可证？

一个新许可证的价格是 **US$125**。许可证在运营者应用中分批发放：当网络中的宿主有 60% 处于忙碌时，会开放新一批许可证和 NANO，这样新增容量随需求增长而加入，已经在服务的宿主也能保持忙碌。

naas 委托

节点即服务（Node-as-a-Service，NaaS）是一种让节点许可证投入工作、而无需自己运行 PC 的方式。提供方为所有者运行这些许可证，并就它所运行的容量收取费用。

为什么要委托

委托适合没有游戏 PC、或者不想把自己的 PC 用于网络的许可证持有者。提供方负责设置、监控和维护。所有者获得这些许可证所服务的每场会话的运营者份额，以美元计价，以 $YOM 支付，再扣除提供方的费用。

一个许可证能赚多少，取决于提供方的表现：把容量放在玩家所在之处并让它保持在线的提供方，会服务更多会话。

常见问题

1. **为什么提供方对网络很重要？** 当高峰需求超出附近家用 GPU 的承载能力时，提供方会在那里增加容量，让这些地区的玩家仍然能得到离自己很近的节点。
2. **我可以改为运行自己的许可证吗？** 可以。许可证一直留在你的账户里，你随时可以用 NANO 在自己的 PC 上运行它。
3. **有费用吗？** 有。提供方就它为你运行的容量收取费用。请先查看提供方的条款。

收益最大化

每场会话都通过协议按统一的比例结算（60% 归运营者，30% 用于网络开发，5% 分给所有继续持有自己收益的人，5% 注销），所以你的收益从不取决于谁卖出了这场会话，也不取决于玩家在哪里。你的份额是每笔会话费用中的固定部分，以美元计价，以 $YOM 支付。你能控制的，是你的机器捕获这股流量的多少。你的收入由两件事相乘而来：**你的硬件真实拥有多少容量，以及它被填满的频率**。本指南涵盖这两方面。每款游戏都会被分配一个会话规格，不由你选择；目前每款游戏都按 Standard 规格运行。

建立你的容量

当宿主的实测资源足以支撑时，它可以同时运行多场会话。它的容量以单位计：一场 Light 会话占用 1 个，Standard 占用 2 个，Premium 占用 4 个，Ultra 占用 6 个，一台机器可以承接任何装得下的组合。rig check 根据 CPU、内存、GPU、编码器和上传能力计算出支持的容量。每个节点许可证解锁一个单位，你的 NANO 包含两个，足够承接一场 Standard 会话。

当你的机器容得下时，再增加容量。容量受第一个耗尽的资源限制：

- **GPU 余量和显存：** 每个规格对每路串流都有固定的预算。
- **编码器会话数和吞吐量：** GPU 必须在其限制之内对每场会话编码。
- **上传带宽：** 估算包含该规格的上传额度和网络余量。
- **CPU、内存和存储：** 完整的游戏、Steam 和串流软件栈必须装得下。

实际的操作顺序：先检查你持续的*上传*速度，然后一次增加一步容量，同时通过仪表板确认每场会话仍然保持其质量。超出你机器真实容量的许可证不会产生收益：网络是按实测能力来分配会话的，而不是按许可证数量。

填满你的容量

HyperOrch 是 YOM 的边缘编排器，它会把每场会话发送到最近的、具备所需 GPU 能力且有空闲容量的节点。因此，要服务更多会话，办法就是在你附近的玩家游玩时，保持在线且有空闲容量。

时间很重要：游戏需求在晚上和周末达到高峰。在这些时段稳定在线的节点，捕获的会话远多于在随机时段在线的节点。

避免突然断线。使用可靠的硬件和稳定的有线连接。如果某个节点没有响应，下一个符合条件的节点会接手这场会话，而持续出故障的节点会被自动隔离，所以持久的优势在于可靠性。

运行它

你通过在自己的 PC 上用 NANO 运行许可证来赚取收益。

特性	NANO
技术知识	极少
设置时间	适中
你要优化的	在你附近的玩家游玩时保持在线
硬件	游戏 PC
运行成本	你自己的电费
承诺	无：随时可以停止。收益取决于电费和当地需求

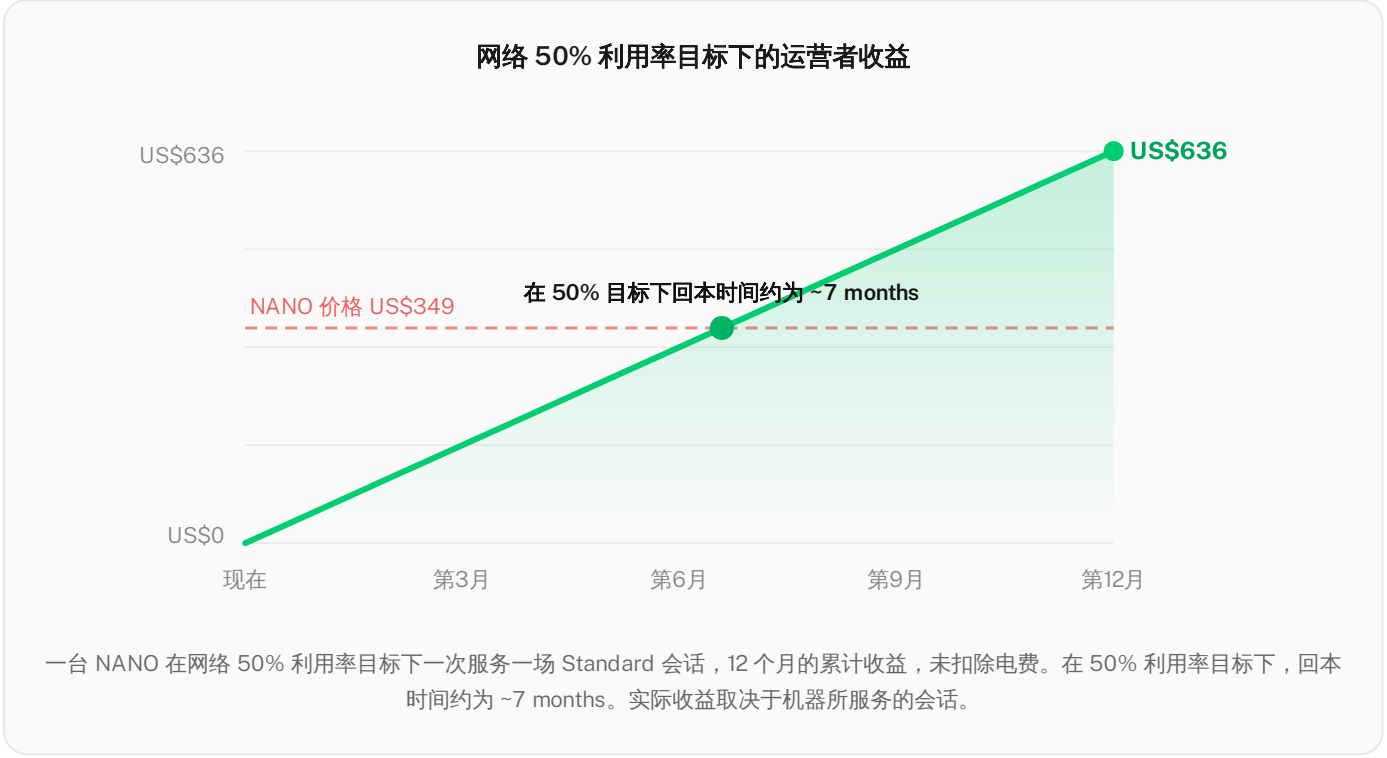
收益预期：实际示例

最常见的问题是“**我能赚多少？**”诚实的回答是一个范围，而不是一个承诺，因为收益随你所在地区的实时网络需求而变化。

网络模型按网络的 **50% 利用率目标**来计算收益：一台机器在其可用的时段里大约有一半时间处于忙碌，会话为典型时长。在这个基础上，一台 NANO（一次一场 Standard 会话）**每月赚取 ~US$53**，每个容量单位约 ~US$26，未扣除电费。你的总额取决于 rig check 在你的宿主上测得的容量（估算见 /operators）。这些是估算，不是保证：网络仍在向这个水平增长，所以今天的收益更低，你的仪表板会显示你自己的会话。

这份收入由两部分组成：**使用量收益**（你从所服务的每笔会话费用中获得的 60% 份额）和**每月奖金**，每台 NANO 每月 US$74，在会话收益之外另行发放。其中 US$37 在机器每天在线至少 6 小时时有保障，另外 US$37 进入一个共享池，按每台机器所服务的会话来分配。它会一直保留，直到更好的补贴机制取代它，我们会提前公布。

回本，基于你已经拥有的硬件： NANO 售价 **US$349**，包含两个许可证，足够承接一场 Standard 会话。在网络 50% 的利用率目标下，一台 NANO 每月赚取 ~US$53，所以在扣除电费之前 ~7 months 回本。新的许可证批次只有在需求填满已经在服务的宿主时才会开放。电费在电价较高的地区影响最大：在目标水平下，一个月的电费在欧洲约为 ~US$12，在大多数其他市场更低，所以请查看你当地的电价。如果你需要先购买一块 GPU，回本时间会更长：这套经济模型是为已经放在你桌子底下的那台机器设计的。



一台 NANO 在网络 50% 利用率目标下一次服务一场 Standard 会话，12 个月的累计收益，未扣除电费。在 50% 利用率目标下，回本时间约为 ~7 months。实际收益取决于机器所服务的会话。

推荐

推荐计划

通过向你的圈子分享 YOM 来赚取奖励。当有人使用你的推荐码购买 YOM NANO（**US$349**，包含两个节点许可证）时，他们能省钱，你能赚取佣金。

如何运作

1. **获取你的推荐码：** 在推荐仪表板中找到你专属的代码
2. **分享它：** 把你的链接发给朋友，发到社交媒体上，或者在社区里分享
3. **赚取奖励：** 当有人使用你的代码购买时，你们双方都受益

奖励

权益	说明
买家折扣	产品价格的一定百分比
推荐人佣金	产品价格（不含增值税）的一定百分比

你的推荐费率可能有所不同。请在你的推荐仪表板中查看你当前的折扣和佣金百分比。

佣金如何计算

佣金按不含增值税的产品价格计算。欧盟标价为 €369，含 21% 增值税（不含增值税为 €305），下面的示例使用这个价格。

示例： 一位买家使用费率为 10% 的推荐码购买了一台 YOM NANO。

步骤	计算	金额
零售价（含 21% 增值税）		€369.00
产品价格（不含增值税）	€369.00 ÷ 1.21	€305.00
买家折扣（10%）	€369.00 × 10%	-€36.90
运费		€18.15
买家支付	€369.00 -€36.90 + €18.15	€350.25
推荐人赚取（10%）	**€305.00 × 10%**	**€30.50**

佣金以不含增值税的产品价格为基础，因为增值税是代税务机关收取的，不属于产品收入。

奖励何时确认

当以下两个条件都满足时，你的佣金即被确认：

- 14 天退货期已过
- 所购买的 NANO 已被激活

你可以在推荐仪表板中追踪待确认和已确认的奖励。

支付

佣金收益在每月月底支付到你已连接的钱包。

分享你的代码

你可以直接分享你的推荐链接，也可以使用仪表板中面向 X (Twitter) 和 Telegram 的快捷分享按钮。

排行榜

排行榜上会显示排名靠前的推荐人。把更多节点运营者带入 YOM 网络，提升你的排名。

分区

构建

概览

YOM 可以把任何编译为 Linux 版本的游戏串流到任何浏览器。没有应用商店，无需下载，对设备也没有要求。玩家点一下链接就能玩。

YOM 为你处理的事项：

- WebRTC 视频/音频串流（H.264、VP8、VP9、Opus）
- 输入捕获：鼠标、键盘、手柄、麦克风
- 自动适配各种设备和屏幕尺寸
- 会话管理和节点分配：YOM 的边缘编排器 HyperOrch 会把每场会话发送到最近的、具备所需 GPU 能力且有空闲容量的节点
- 通过 Google Tag Manager 集成提供分析

在 YOM 上构建的三种方式

选择适合你需求的路径：

路径	你需要什么	上线所需时间	是否需要 SDK？
快速开始	把游戏编译为 Linux 版本，上传，嵌入 iframe	约 15 分钟	不需要
Web 集成	自定义界面、多路串流页面、网页到游戏的命令	约 1 小时	JS 嵌入脚本
你自己的串流器或模型	你的串流器或语音模型，经审核并固定版本，然后作为托管会话运行	合作伙伴集成，从范围明确的试点开始	请联系我们

快速开始：串流任何游戏

上传你编译好的 Linux 二进制文件，再嵌入一个 iframe。其余一切由 YOM 的通用串流器处理：WebRTC、输入、视频、音频。无需改动你的游戏代码。

最适合： 演示、展示、单人体验、快速原型。

Web 集成：自定义门户

使用 YOM 嵌入脚本（`yom-embed.js`）构建自定义游戏门户，支持多路串流、动态布局、网页与游戏之间的通信，以及带品牌的引导流程。

最适合： 游戏门户、内容聚合站、带品牌的店面、互动营销页面。

你自己的串流器或模型：托管会话

为自己的用户串流游戏的平台，或运行自有语音模型的公司，可以把它作为托管会话部署在 YOM 上。YOM 会审核你的串流器或模型，并固定已审核的版本；此后它只通过会话路径运行。HyperOrch 把每场会话安排到靠近用户的节点上，NodeOS 在它自己的容器里运行它，这个容器为该会话创建，会话关闭时即被丢弃。用户关系和品牌归你所有。

平台以 **YOM Games** 的名义购买，语音团队以 **YOM AI** 的名义购买，两者都按会话计费，使用相同的算力规格，每场会话 US$0.04 到 US$0.24。

最适合： 运行自有游戏串流器的串流平台，以及拥有自有语音和语言模型的公司。

每种路径如何销售

快速开始和 Web 集成接入的是 **YOM Games** 按会话销售的方式：通过 Walter（YOM 自己的消费者服务），或通过你自己的渠道。上面的操作方法是一样的；不同的是你如何走向市场，以及如何计费。

- **Discovery** 为免费的嵌入式演示提供支持。用快速开始或 Web 集成，把一个可玩的演示放到 Reddit、Discord、Telegram、落地页或广告位上。不收服务费；会话按基础算力成本运行。这是把一次点击变成一位玩家的切入点。
- **Hosting Slot** 是面向发行商的方案，**每月 US$89**：2,300 积分（1,150 场 Standard 会话）和一个工作负载，超出部分按每积分 US$0.048 计。付费方案从范围明确的试点或 Hosting Slot 开始。大多数工作室在这里使用快速开始。
- **Compute Pass** 是 Walter 的消费者订阅（2027 年 1 月起计费）：每月 US$4.99 / US$9.99 / US$19.99，分别含 30 / 70 / 140 积分。每场会话在 Light 规格消耗 1 积分，Standard 消耗 2，Premium 消耗 4，Ultra 消耗 6，最长 60 minutes。每月额度可以覆盖不同规格会话的组合；额外积分每个 US$0.175。玩家关联 Steam 或 Epic，串流自己已经拥有的游戏（自带游戏）。发行商把自己的玩家群引导到 Compute Pass，把它作为串流层，同时保留自己的商店和收入。

关于每月额度与会话规格和游玩时长的关系，参见 Compute Pass 积分。

无论哪种方式适合你，本页的集成工作都是通往它的路径：YOM 是基础设施，不是抽成的商店。变现方式由你掌控，无论是订阅、付费大作、广告支持的演示，还是免费展示。

为即点即玩而设计

云串流游戏位于网页上，在任何设备上都能立即访问。要充分发挥即点即玩的优势：

- **跳过主菜单：** 玩家希望直接进入游戏。
- **响应式设计：** 你的游戏会在手机、平板和桌面设备上运行。
- **以网页原生的方式思考：** 在合适的地方，用网页小组件和游戏与网页之间的双向通信，代替游戏内界面。

快速开始

两步让游戏在 YOM 上串流：上传你的二进制文件，然后嵌入一个 iframe。

> **网络状态：** 生产网络自 2026 年 6 月 5 日起一直在线运行。下面的快速开始流程今天即可使用。随着网络扩大，如果仪表板中的某个步骤看起来有所不同，请通过 info@yom.net 联系我们。

步骤 1：部署你的串流

1. 把你的游戏编译为 **Linux x86_64**。任何引擎都可以：Unreal Engine、Unity、Godot 或自研引擎。
2. 登录 app.yom.net/streams。
3. 创建一个新的串流：
   - **Stream Name**：唯一标识符（例如 `my-game-demo`）
   - **Build Target**：选择你的目标环境（例如 Development）
   - **Upload Binaries**：把编译好的游戏文件拖放进来
   - **Choose Network**：选择你偏好的网络（例如 Test Net）
4. 检查你的配置，然后点击 **Pay + Deploy**。
5. 等待服务端和客户端文件都上传完成。上传期间不要刷新或离开页面。

部署完成后，你会收到一个 **Game ID** 和一个 **Stream ID**。

步骤 2：嵌入串流

把这个 iframe 添加到你的网页：

```
<iframe
  src="https://player.yom.net?gameId=YOUR_GAME_ID&streamId=YOUR_STREAM_ID&origin=https://yoursite.com"
  width="1280"
  height="720"
  frameborder="0"
  allowfullscreen
  allow="autoplay *; microphone *; clipboard-write *; pointer-lock"
  style="border: none;">
</iframe>
```

把 `YOUR_GAME_ID`、`YOUR_STREAM_ID` 和 `https://yoursite.com` 替换为你的实际值。

就这么简单。玩家访问你的页面，点击播放，游戏就直接串流到他们的浏览器。

可选的 iframe 参数

把这些作为查询参数追加到 iframe 的 `src` URL 后面：

参数	取值	说明
`guestName`	字符串	预先填入玩家名称
`requestMic`	`true` / `false`	请求麦克风权限
`forceMic`	`true` / `false`	必须开启麦克风才能继续
`fullscreen`	`none` / `window` / `native`	全屏模式（`window` 在 iOS 上可用）
`autoInitiate`	`true` / `false`	跳过播放按钮，立即开始
`requiresName`	`true` / `false`	要求输入玩家名称（默认：true）
`lang`	ISO-639-1 代码	语言代码（例如 `eng`）
`publisher`	字符串	用于分析的发行商名称
`coppa`	`0` / `1`	COPPA 合规标志
`genre`	字符串	用于广告定向的游戏类型

安全：CSP 标头

如果你的网站使用 Content Security Policy 标头，请允许 YOM 播放器的来源：

```
Content-Security-Policy: frame-src 'self' https://player.yom.net;
```

或通过 meta 标签：

```
<meta http-equiv="Content-Security-Policy" content="frame-src 'self' https://player.yom.net;">
```

故障排查

如果遇到问题：

1. 检查浏览器控制台中的错误信息。
2. 确认你的 `gameId` 和 `streamId` 正确无误。
3. 确保你的 CSP 标头允许 `frame-src https://player.yom.net`。
4. 确认你的游戏二进制文件是为 Linux x86_64 编译的。

接下来做什么？

- **Web 集成：** 使用 YOM 嵌入脚本添加自定义界面、多路串流页面和网页到游戏的通信。

如需协助，请通过 info@yom.net 联系我们。

js 集成

YOM 嵌入脚本（`yom-embed.js`）让你完全掌控串流体验。它管理 iframe 的创建、播放器的生命周期事件、全屏行为、可见性切换，以及你的网页与游戏之间的双向通信。

入门

在你的页面中引入嵌入脚本：

```
<script src="https://cdn.yom.net/yom-embed.js"></script>
```

然后添加所需的 HTML 元素：

```
<!-- Player container: the iframe will be injected here -->
<div class="yom-player-placeholder"
     data-yom-stream-id="1"
     data-game-id="YOUR_GAME_ID"
     style="width: 100%; aspect-ratio: 16/9;">
</div>

<!-- Launch button -->
<input class="yom-name-input" data-yom-stream-id="1" placeholder="Your name">
<button class="yom-launch-button" data-yom-stream-id="1">Play</button>
```

当玩家点击按钮时，`yom-embed.js` 会在占位 div 里创建一个 iframe，连接到 YOM，并开始游戏串流。

核心概念

必需的 CSS 类

类	用途
`.yom-player-placeholder`	创建串流 iframe 的容器
`.yom-launch-button`	发起串流的按钮
`.yom-name-input`	可选的玩家名称文本输入框

必需的 data 属性

属性	用于	说明
`data-yom-stream-id`	所有元素	把元素关联到某个特定的串流实例
`data-game-id`	`.yom-player-placeholder`	你的游戏的唯一标识符

可选的 data 属性

把这些设置在 `.yom-player-placeholder` 上，用来配置行为：

属性	取值	说明
`data-request-mic`	`true` / `false`	请求麦克风权限
`data-force-mic`	`true` / `false`	必须开启麦克风才能继续
`data-fullscreen`	`none` / `window` / `native`	全屏模式（默认：`none`）
`data-auto-initiate`	`true` / `false`	无需点击按钮，自动开始
`data-require-name`	`true` / `false`	要求输入名称（默认：`true`）
`data-hook`	CSS 选择器	串流期间给该元素添加 `.yom-active` 类
`data-callback`	函数名	启动前的自定义验证
`data-lang`	ISO-639-1	语言代码
`data-publisher`	字符串	发行商名称
`data-coppa`	`0` / `1`	COPPA 合规
`data-genre`	字符串	游戏类型

完整示例

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="Content-Security-Policy"
        content="frame-src 'self' https://player.yom.net;">
  <style>
    .yom-toggle-visibility { transition: opacity 0.3s ease; }
  </style>
</head>
<body>
  <!-- Stream container -->
  <div class="yom-player-placeholder"
       data-yom-stream-id="demo"
       data-game-id="my-game"
       data-fullscreen="window"
       style="width: 100%; aspect-ratio: 16/9;">
  </div>

  <!-- Pre-stream UI (hidden when stream starts) -->
  <div class="yom-toggle-visibility" data-yom-stream-id="demo">
    <input class="yom-name-input" data-yom-stream-id="demo" placeholder="Enter your name">
    <button class="yom-launch-button" data-yom-stream-id="demo">Start Game</button>
  </div>

  <!-- In-stream controls (shown when stream starts) -->
  <div class="yom-toggle-visibility" data-yom-stream-id="demo" style="display: none;">
    <button onclick="window.postToPlayer('demo', 'SpawnBot', {amount: '1'})">Add Bot</button>
    <button onclick="window.postToPlayer('demo', 'ClearBot', {})">Clear Bots</button>
  </div>

  <script src="https://cdn.yom.net/yom-embed.js"></script>
</body>
</html>
```

接下来做什么？

- **自定义身份验证**： 在串流启动前添加验证
- **事件处理**： 向游戏发送命令并监听游戏事件
- **动态布局**： 根据串流状态切换界面可见性
- **简易界面钩子**： 全屏模式、自动缩放和容器钩子
- **多路串流设置**： 在一个页面上运行多路游戏串流

自定义身份验证

你可以在串流启动前运行自定义验证，例如检查密码、验证令牌，或者让访问取决于一次表单提交。

设置

在启动按钮上添加 `data-callback`，值为你的验证函数的名称：

```
<div class="yom-player-placeholder"
     data-yom-stream-id="1"
     data-game-id="my-game"
     style="width: 100%; aspect-ratio: 16/9;">
</div>

<input type="text" class="yom-name-input" data-yom-stream-id="1" placeholder="Your name">
<input type="password" id="password" placeholder="Enter password">
<button class="yom-launch-button"
        data-yom-stream-id="1"
        data-callback="validatePassword">
  Start Game
</button>
```

定义回调函数

你的函数会收到两个参数：`onSuccess` 和 `onFailure`。根据验证结果调用其中一个。

```
function validatePassword(onSuccess, onFailure) {
    const password = document.getElementById('password').value;

    if (password === 'correct-password') {
        onSuccess();   // Stream launches
    } else {
        onFailure();   // Launch is cancelled, button resets
    }
}
```

回调会在玩家点击启动按钮时被调用，**早于** iframe 的创建。如果该函数抛出错误，启动会被取消，错误会被记录。

使用场景

- **密码限制访问：** 只允许持有代码的用户使用串流
- **表单验证：** 要求在启动前填完必填字段
- **外部身份验证：** 调用你自己的 API 来验证用户，然后在响应处理函数中调用 `onSuccess()`
- **年龄验证：** 在允许游玩之前提示确认年龄

事件处理

YOM 支持你的网页与游戏串流之间的双向通信。向游戏发送命令，并监听传回的事件。

向游戏发送命令

使用 `window.postToPlayer()` 向正在运行的串流发送命令：

```
window.postToPlayer(streamId, commandName, data);
```

参数	类型	说明
`streamId`	字符串	必须与目标串流的 `data-yom-stream-id` 一致
`commandName`	字符串	在你的游戏中注册的函数名
`data`	对象	传给游戏函数的键值对

示例：机器人控制

```
<button onclick="window.postToPlayer('1', 'SpawnBot', {amount: '1'})">Add Bot</button>
<button onclick="window.postToPlayer('1', 'RemoveBot', {amount: '1'})">Remove Bot</button>
<button onclick="window.postToPlayer('1', 'ClearBot', {})">Clear All Bots</button>
```

示例：镜头设置

```
window.postToPlayer('1', 'SetCameraSensitivityX', { MouseSensitivityX: '0.5' });
window.postToPlayer('1', 'SetCameraSensitivityY', { MouseSensitivityY: '0.5' });
```

示例：串流控制

```
window.postToPlayer('1', 'pause', {});
window.postToPlayer('1', 'resume', {});
```

监听来自游戏的事件

YOM 播放器通过 `postMessage` 向你的页面发送事件。在 `window` 对象上监听它们：

```
window.addEventListener('message', function(event) {
    // Only trust messages from the YOM player
    if (event.origin !== 'https://player.yom.net') return;

    const { event: eventName, streamId, data } = event.data;

    switch (eventName) {
        case 'playerReady':
            console.log(`Stream ${streamId} is ready`);
            break;
        case 'streamActive':
            console.log(`Stream ${streamId} is now playing`);
            break;
        case 'streamClosed':
            console.log(`Stream ${streamId} ended`);
            break;
        case 'error':
            console.error(`Stream ${streamId} error:`, data.message);
            break;
    }
});
```

可用事件

事件	触发时机
`playerReady`	串流 iframe 已加载并就绪
`streamActive`	游戏已开始，玩家已进入体验
`streamClosed`	玩家退出或串流结束
`error`	出了问题
`requestFullscreen`	玩家请求全屏
`exitFullscreen`	玩家退出全屏

Google Tag Manager 集成

播放器的所有事件都会自动推送到 `window.dataLayer`，用于 GTM/GA 集成：

```
// Automatically pushed by yom-embed.js
window.dataLayer.push({
    'event': 'streamActive',
    'category': 'yom-event',
    'streamId': '1',
    'sessionId': 'abc123'
});
```

配置你的 GTM 容器，捕获 `yom-event` 类别的事件，用于分析追踪。

动态布局

根据串流是否处于活动状态，显示或隐藏页面元素。这样你可以构建在游戏开始时消失的串流前落地页界面，以及在游戏进行时出现的串流中控件。

可见性切换

给任何你想切换的元素添加 `.yom-toggle-visibility` 类和 `data-yom-stream-id`：

```
<!-- Visible before stream, hidden during stream -->
<div class="yom-toggle-visibility" data-yom-stream-id="1">
    <h2>Welcome! Click below to start playing.</h2>
    <button class="yom-launch-button" data-yom-stream-id="1">Play Now</button>
</div>

<!-- Hidden before stream, visible during stream -->
<div class="yom-toggle-visibility" data-yom-stream-id="1" style="display: none;">
    <p>Game is running. Use the controls below:</p>
    <button onclick="window.postToPlayer('1', 'SpawnBot', {amount: '1'})">Add Bot</button>
</div>
```

工作方式：

1. 当串流变为活动状态（`playerReady`）时，元素会切换：可见的淡出，隐藏的淡入。
2. 当串流关闭（`streamClosed`）时，元素会切换回原来的状态。

CSS 过渡

淡入淡出动画为 300ms。添加下面的 CSS，让过渡更平滑：

```
.yom-toggle-visibility {
    transition: opacity 0.3s ease;
}
```

容器钩子

在 `.yom-player-placeholder` 上使用 `data-hook`，在串流运行时给任何容器元素添加 `.yom-active` 类：

```
<div id="game-section">
    <div class="yom-player-placeholder"
         data-yom-stream-id="1"
         data-game-id="my-game"
         data-hook="#game-section"
         style="width: 100%; aspect-ratio: 16/9;">
    </div>
</div>
```

现在你可以用 CSS 为活动状态设置样式：

```css
#game-section.yom-active {
    background: #000;
    padding: 0;
    max-width: 100%;
}
```

`.yom-active` 类在串流开始时添加，在串流结束时移除，让你完全用 CSS 控制布局变化。

简易界面钩子

YOM 嵌入脚本会自动处理全屏管理、宽高比缩放和串流生命周期。下面介绍如何配置每一种行为。

全屏模式

通过 `.yom-player-placeholder` 上的 `data-fullscreen` 设置全屏模式：

```
<div class="yom-player-placeholder"
     data-yom-stream-id="1"
     data-game-id="my-game"
     data-fullscreen="window">
</div>
```

模式	行为	iOS 支持
`none`	不使用全屏（默认）	不适用
`window`	基于 CSS 的全屏（`position: fixed`，铺满视口）	支持
`native`	Fullscreen API（真正的全屏，隐藏浏览器界面）	不支持（回退为 `window`）

Window 模式推荐用于跨平台支持。它在 iOS 上可用（iOS 上没有原生的 Fullscreen API），并且能处理祖先元素的 CSS 变换，否则这些变换会破坏 `position: fixed`。

全屏切换循环

当玩家反复请求全屏时，模式会循环切换：

- 桌面端：`none` → `window` → `native` → `none`
- iOS：`none` → `window` → `none`（跳过 native）

自动缩放

串流会自动缩放以填满它的容器，同时保持正确的宽高比。当全屏模式改变或窗口大小调整时，嵌入脚本会把新的视口尺寸通知播放器，游戏分辨率随之调整。

无需配置。开箱即用。

串流生命周期

当串流处于活动状态时，嵌入脚本会自动管理以下几件事：

1. **创建 iframe：** 启动时在 `.yom-player-placeholder` 内创建，并带有正确的权限（`autoplay`、`microphone`、`clipboard-write`、`pointer-lock`）。
2. **透明度淡入：** iframe 就绪后在 300ms 内淡入。
3. **可见性切换：** 带有 `.yom-toggle-visibility` 的元素会在串流开始和关闭时被切换。
4. **容器钩子：** `data-hook` 所指向的元素会获得 `.yom-active` 类。
5. **视口通知：** 调整大小或全屏变化时，会把新的尺寸通知播放器。

自动启动

完全跳过启动按钮，页面一加载就开始串流：

```
<div class="yom-player-placeholder"
     data-yom-stream-id="1"
     data-game-id="my-game"
     data-auto-initiate="true">
</div>
```

玩家名称可以通过 `guestName` iframe 参数或 `.yom-name-input` 字段预先填入。

可观测性

嵌入脚本内置了 Grafana Faro 集成，用于真实用户监控。控制台日志、错误和追踪会被自动捕获，你这边无需任何设置。

错误也会上报到 YOM 后端，用于调试。

多路串流设置

YOM 支持在单个页面上运行多路相互独立的游戏串流。每一路串流都有自己的容器、启动按钮和通信通道。

设置

给每一路串流一个唯一的 `data-yom-stream-id`：

```
<!-- Stream 1 -->
<div class="yom-player-placeholder"
     data-yom-stream-id="1"
     data-game-id="racing-game"
     style="width: 100%; aspect-ratio: 16/9;">
</div>
<input class="yom-name-input" data-yom-stream-id="1" placeholder="Name for Racing">
<button class="yom-launch-button" data-yom-stream-id="1">Start Racing</button>

<!-- Stream 2 -->
<div class="yom-player-placeholder"
     data-yom-stream-id="2"
     data-game-id="fps-game"
     style="width: 100%; aspect-ratio: 16/9;">
</div>
<input class="yom-name-input" data-yom-stream-id="2" placeholder="Name for FPS">
<button class="yom-launch-button" data-yom-stream-id="2">Start FPS</button>
```

每一路串流独立运行，分别启动、停止和通信。

向指定串流发送命令

使用 `window.postToPlayer()`，并传入目标串流的 ID：

```
// Send command to stream 1
window.postToPlayer('1', 'SpawnBot', { amount: '1' });

// Send command to stream 2
window.postToPlayer('2', 'SetCameraSensitivityX', { MouseSensitivityX: '0.3' });
```

监听来自指定串流的事件

来自播放器的事件带有 `streamId`，因此你可以区分它们来自哪一路串流：

```
window.addEventListener('message', function(event) {
    if (event.origin !== 'https://player.yom.net') return;

    const { event: eventName, streamId } = event.data;

    if (eventName === 'streamActive') {
        console.log(`Stream ${streamId} started playing`);
    }
});
```

按串流划分的可见性切换

可见性切换的作用范围限定在各自的 `data-yom-stream-id`：

```
<!-- Only toggles when stream 1 changes state -->
<div class="yom-toggle-visibility" data-yom-stream-id="1">
    Controls for Racing Game
</div>

<!-- Only toggles when stream 2 changes state -->
<div class="yom-toggle-visibility" data-yom-stream-id="2">
    Controls for FPS Game
</div>
```

使用场景

- **游戏门户：** 在单个落地页上展示多款游戏
- **对比页面：** 让玩家并排试玩两种体验
- **内容聚合：** 汇集来自不同发行商的串流
- **活动：** 在一个仪表板上运行多路赛事串流

分区

关于

代币分配

属性	数值
代币总量	750,000,000
智能合约	0xb6314518b61b4864162c7aE7fdc36261e0A14C4b
主链	Avalanche C-Chain
结算链	Avalanche C-Chain
上线时间	2026 年 6 月 5 日

每笔会话费用的去向

每笔会话费用都按固定条款分配：60% 归服务该会话的 GPU 所属的运营者，30% 用于网络开发，5% 分给所有继续持有自己收益的人，5% 永久注销。

代币分配

类别	分配比例	代币数量	解锁时间表
国库	37.50%	281,250,000	TGE 时解锁 20%，9 个月悬崖期，36 个月线性解锁
生态	25.00%	187,500,000	TGE 时解锁 20%，6 个月悬崖期，24 个月线性解锁
HODL	17.31%	129,790,869	9 个月悬崖期，16 个月线性解锁
流动性池	9.72%	72,934,111	TGE 时解锁 35%，24 个月线性解锁
种子轮	2.00%	15,000,000	3 个月悬崖期，悬崖期结束时解锁 10%，6 个月线性解锁
私募轮	1.50%	11,250,000	3 个月悬崖期，12 个月线性归属
KOL 轮	0.48%	3,571,429	TGE 时解锁 20%，3 个月悬崖期，6 个月线性解锁
团队	3.00%	22,500,000	9 个月悬崖期，20 个月线性解锁
社区	2.78%	20,860,735	3 个月悬崖期，5 个月线性解锁
交易所	0.03%	200,000	TGE 时解锁 100%
社区轮 (i)	0.10%	714,286	TGE 时解锁 10%，3 个月悬崖期，6 个月线性解锁
社区轮 (ii)	0.10%	714,286	TGE 时解锁 10%，3 个月悬崖期，6 个月线性解锁
社区轮 (iii)	0.10%	714,286	TGE 时解锁 10%，3 个月悬崖期，6 个月线性解锁
任务	0.40%	3,000,000	3 个月悬崖期，2 个月线性解锁

- **国库（37.50%）**：推动采用、交易所上线、工作室资助和日常运营。
- **生态（25.00%）**：专为节点运营者保留：启动期奖励（随着会话付费足以覆盖运营者而逐步减少），以及空投、大使和资助计划，还有流动性激励。

- **HODL（17.31%）**：承诺长期持有的忠实持有者，按延长的归属时间表发放。
- **流动性池（9.72%）**：专用于 DEX 和交易所的流动性。确保在中心化和去中心化的交易场所都有深度充足的交易池。
- **种子轮（2.00%）**：从第一天起就参与的机构股权和代币认股权证投资者，以及种子代币投资者。3 个月悬崖期，悬崖期结束时解锁 10%，其余部分在 6 个月内线性归属。
- **私募轮（1.50%）**：采用延长归属期的私募代币投资者。3 个月悬崖期，悬崖期结束时不解锁；全部余额在 12 个月内线性归属。
- **KOL 轮（0.48%）**：面向受聘为代币上线营销提供触达的网红和 KOL 合作伙伴的关键意见领袖推广活动。TGE 时解锁 20%，用于配合上线推广；其余 80% 在 3 个月悬崖期之后，于 6 个月内线性归属。
- **团队（3.00%）**：分配给所有核心团队成员。在较长的时间跨度内归属，以确保与网络的成功深度一致。
- **社区（2.78%）**：早期就相信这个项目的社区投资者。这一部分代表我们的 OG 持有者，他们选择了持有并桥接。
- **交易所（0.03%）**：为上线合作伙伴和交易所上市保留的剩余部分。
- **社区轮（i / ii / iii）（每轮 0.10%，合计 0.30%）**：三个独立的社区轮批次，每个合作平台一个，于 2026 年 5 月开放。TGE 时归属 10%，然后是 3 个月悬崖期和 6 个月线性归属。
- **任务（0.40%）**：驱动排行榜竞赛和网红合作。支持带动认知度和采用的营销合作。

路线图

YOM 正在构建会话边缘：在每位玩家和每位 AI 用户身边提供 GPU 容量，按会话销售，使用的是玩家和电信运营商已经拥有的 GPU。运营者从第一天起就在服务真实的会话，网络随着它所服务的会话一起成长。

YOM 运行的内容

已在 YOM 网络上交付并运行：

网络与代币

- **生产网络**：网络每天都在服务实时会话
- **$YOM 代币**：运营者以 $YOM 获得报酬，代币已在 Avalanche 上线。完整经过见 TGE 更新和战略更新
- **每场会话的费用分配**：60% 归服务该会话的运营者，30% 用于网络开发，5% 分给所有继续持有自己收益的人，5% 注销
- **以美元计价的运营者报酬**：每笔会话费用中的固定份额，以美元计价，以 $YOM 支付

串流与游戏

- 在任何设备上即时通过浏览器游玩你拥有的 Steam 和 Epic 游戏，无需下载
- 通用串流器（Universal Streamer），支持 WebRTC 视频和音频以及完整的输入（鼠标、键盘、手柄、麦克风）
- Skyhook Games 的 Roughneck Rumble 在网络上串流

节点网络

- **HyperOrch**，YOM 的边缘编排器，把每场会话发送到最近的、具备所需 GPU 能力且有空闲容量的节点
- **容器化节点软件栈**：节点软件以容器的形式运行在合作伙伴自己的服务器和电信机架上，与家用 GPU 使用同一套软件
- **英国电信边缘**：在英国最大的移动通信塔公司 Cornerstone 的铁塔站点上，与 StonesThro 和 Zadara 一起，在 20 天内运行了 168 场实时游戏会话
- **会话结果记录**，以及节点版本的自动化测试周期
- NANO：插到游戏 PC 上，扫描二维码，这台 PC 就可以开始服务会话
- 为现场的 NANO 设备提供无线（OTA）更新
- 向节点运营者按会话付款
- 通过驱动程序的时间切片，让多场会话共享一块 GPU
- 硬件配置扫描和自动验证
- 节点和会话的日志记录与监控

运营者工具

- app.yom.net 运营者门户
- stats.yom.net 公开的网络统计
- 带佣金追踪的推荐计划

安全

- 对 NANO 节点镜像的外部渗透测试
- 每场会话都在自己干净的容器中运行，会话关闭时即被丢弃
- NANO 从它自己的加密驱动器启动一个只读系统，该驱动器只能通过 YOM 的密钥服务解锁

- NANO 节点在每次启动时都会对照已知哈希检查自己的启动文件，未通过的节点会被自动隔离
- 串流在传输中加密（WebRTC、DTLS-SRTP）
- 受保护的后端 API（限流、认证加固、输入验证）

我们的方法

YOM 的路线图遵循三项原则：

1. **先有经济性，再放量。** 每个产品先和合作伙伴在小规模上证明它的价格，然后再推动规模。
2. **每项能力都由 Walter 先行。** Walter 是 YOM 自己的消费者服务，每项新能力都先在 Walter 的实时会话中运行，然后才卖给合作伙伴。
3. **供给和市场在关口处开放。** 新的 NANO 批次和新的市场，依据实测的使用情况和经过测试的容量开放。插上 NANO，它就可以开始服务会话；收益随它所服务的会话而来。

社交媒体

关注这个网络：发布、实时数据和运营者动态。每个渠道都有各自的内容和社区，下面是 YOM 的官方账号。

- X
- Telegram
- Discord
- YouTube
- Medium
- LinkedIn
- Instagram

常见问题

1. 常规问题

YOM 是什么？

YOM 把 GPU 放在每位用户身边：这是一个去中心化的边缘网络，把闲置的消费级 GPU 和电信机架变成一张算力网格，无需提前预订。YOM 销售托管的实时会话，一个网络上的两种产品，都按会话付费：**YOM Games**，面向玩家的实时游戏串流（通过 Walter），也面向发行商，以及向自己的用户串流游戏的平台；**YOM AI**，实时 AI 会话，语音优先，面向运行自有语音和语言模型的公司。YOM 把来自自托管节点和委托节点的闲置 GPU 资源，以远低于中心化云的成本，连接给工作室、品牌和广播机构。

YOM 如何运作？

YOM 运行在由游戏电脑（节点）组成的网络上，这些节点运行我们自研的 **Node OS**，向终端用户串流会话。每场会话都在 YOM 的托管软件栈中、在它自己的容器里运行，该容器为这场会话创建，会话关闭时即被丢弃。玩家、发行商、平台和 AI 公司按会话付费。服务这场会话的 GPU 所属的运营者获得其费用的 60%：这是一个固定份额，以美元计价，以 $YOM 支付。

谁能从 YOM 中受益？

- **节点运营者：** 你的 GPU 服务的每场会话，你都获得 60%。
- **社区成员：** 赚取 XP，继续持有自己的收益以分享社区池，并在代币持有者投票开放后，对供应计划投票。
- **工作室与平台：** 用 YOM Games 把你的游戏串流到任何浏览器，紧挨着你的玩家，成本远低于数据中心。
- **AI 公司：** 用 YOM AI 在紧挨着说话者的实时会话中运行你自己的语音模型。

2. 节点运营者

我如何开始运行 YOM 节点？

1. **自托管（现已可用）：**
   - 把 **NANO** 插到你的游戏 PC 上，或者通过**手动安装**在你的游戏电脑上设置节点。
2. **额外的许可证：**
   - 许可证让你机器的更多容量投入服务。一个新许可证是 US$125。

自托管的硬件要求是什么？

- **GPU：** 任何 NVIDIA RTX 配置（2060RTX 及以上），高显存有助于最大化节点容量。
- **CPU：** Intel i5/i7 或 AMD Ryzen 7/9，多任务处理和稳定性更好。
- **内存：** 低配置的设备 16GB 就够了，但推荐 32GB 以上。
- **存储：** 即插即用方案无需存储。自定义 Linux 启动部署：需要 1TB 以上的可用空间。
- **网络：** 有线以太网连接，实测上传速度至少 20 Mbps。

运行节点能赚多少？

收益取决于你的节点所服务的会话。在网络的 50% 利用率目标下：

- **每台 NANO：** 每月约 ~US$53，未扣除电费。
- **每个宿主：** 根据你的硬件的实测容量计算。运行免费的 PC 检测，查看你的结果。
- **回本：** NANO 约 ~7 months。实时数字见 /operators，并会随每次模型发布更新。

一台 NANO 是 US$349，包含 2 个许可证，足够承接一场 Standard 会话；每个新许可证（US$125）解锁一个单位的容量，即一场 Light 会话的大小。运营者的报酬是每笔会话费用的固定份额，以美元计价，以 $YOM 支付，并随你的节点所服务的会话而增长。这些是估算，不是保证：网络仍在向它的目标增长，所以今天的收益更低。容量和许可证如何运作：节点许可证。

自托管时应该考虑哪些成本？

成本包括：

- **硬件：** 一台新的整机，或者你已经拥有的游戏 PC。
- **电费：** 你的 GPU 在服务会话时消耗的电力。
- **许可证：** 每个节点许可证让你机器的容量中多一个单位投入服务。

---

3. 工作室与平台

为什么应该用 YOM 做云游戏？

YOM 提供：

- **成本效率：** 按会话付费，按算力规格从 US$0.04 到 US$0.24：一场最长一小时的 Standard 会话为 US$0.08，而同样一小时在为高峰预留的数据中心 GPU 上约为 US$1.23。
- **可扩展性：** 使用不断增长的节点网络，无需基础设施投入。
- **易于集成：** 用 iframe 把串流直接嵌入网站、应用和平台。
- **你的品牌：** 以你自己的品牌呈现串流。

如何确保不同设备上的游戏性能？

YOM 的边缘编排器 HyperOrch 会把每场会话发送到最近的、具备所需 GPU 能力且有空闲容量的节点。距节点 50 km 以内的玩家，看到的单向延迟中位数约为 15 ms；目标是 10 ms 以内。

YOM 采用什么定价模式？

YOM 对**每一场会话**都按算力规格固定定价，YOM Games 和 YOM AI 都一样：

- Light / Standard / Premium / Ultra 会话，**每场会话分别为 US$0.04 / US$0.08 / US$0.16 / US$0.24**，无论是游戏还是实时 AI 对话；一场会话最长一小时。
- **Hosting Slot，每月 US$89：** 2,300 积分（1,150 场 Standard 会话）和一个工作负载；超出部分每积分 US$0.048。

会话可以由广告支持吗？

可玩广告和广告支持的会话在路线图上，今天尚未上线。为免费演示（Discovery）提供支持的同一套可嵌入串流，可以承载会话前广告或可玩广告形式，等网络密度足以支撑时，我们会扩展广告支持的模式。在此之前，发行商按会话付费或通过 Hosting Slot 付费，Walter 订阅（Compute Pass）从 2027 年 1 月起计费。

---

4. 代币经济学与奖励

$YOM 是什么，它如何运作？

$YOM 结算网络上的每一场会话：

- **运营者报酬：** 运营者以 $YOM 获得报酬，这是每笔会话费用中的一个固定份额，以美元计价。
- **社区份额：** 每笔会话费用的 5% 分给所有继续持有自己收益的人。
- **注销：** 每笔会话费用的 5% 永久注销。
- **流动性：** $YOM 的主要市场在 Avalanche 上。

XP 是什么，它在生态中起什么作用？

XP 是通过为网络做贡献而获得的**不可转让的贡献分数**。它用于：

- 在排行榜上为贡献者排名。
- 在投票开放后，为代币持有者对供应计划的投票加权。

XP 不会改变你的节点服务哪些会话，也不会改变它的收益。

$YOM 的供应量会增长吗？

不会。除了固定的总供应量 750,000,000 $YOM 之外，不会再铸造新代币，并且每笔会话费用的 5% 会被永久注销。

我如何赚得更多？

收益随你的节点所服务的会话而变化：

- 保持你的节点在线且可靠，使用有线连接。
- 机器越强，容量越大，免费的 rig check 会测出它。
- 每增加一个节点许可证，就多让一个单位的容量投入工作。

---

5. 治理与社区

YOM 生态中的治理如何运作？

代币持有者对供应计划的投票是治理的下一步（见路线图）。开放后，持有者将对以下事项投票：

- 运营者激励费率。
- 国库分配。
- 硬件标准。
- 资助的批准。

今天由谁治理协议，这又将如何改变？

今天，一个由三名创始人代表组成的理事会运行网络的结算合约，并设定供应计划参数、国库分配和资助。接下来是代币持有者投票，每票的权重 67% 由 XP 决定，33% 由 $YOM 持仓决定，这样，做事的运营者也能帮助引导网络。之后，社区选举产生的理事会成员开始接替创始人代表。

我如何参与 YOM 社区？

你可以通过以下方式参与 YOM：

- 在 **Discord 和 Telegram** 上参与讨论。
- 赚取 XP，它会在代币持有者投票开放时为你的投票加权。
- 举办社区活动，吸引新用户加入并扩大生态。

---

6. 技术与故障排查

如果我的节点没有出现在仪表板上，该怎么办？

- 确保你的钱包已正确连接。
- 检查你的网络连接，并重启系统。
- 如果问题仍然存在，请联系 YOM 支持。

我如何优化我的设备以获得更好的性能？

- 确保良好的散热和电源效率。
- 使用高速有线网络连接。
- 通过仪表板定期监控性能指标。

compute pass 积分说明

2026 年 9 月 17 日

Walter 已公布的 Compute Pass 定价采用每月积分额度。每场会话按其算力规格消耗积分：

会话规格	每场会话消耗的积分
Light	1
Standard	2
Premium	4
Ultra	6

这更正了此前关于一个积分可以购买任何一场会话的说法。一个积分可以覆盖一场 Light 会话；Standard、Premium 和 Ultra 消耗更多。玩家可以把每月额度用在不同规格的组合上。

已公布的订阅方案保持不变：US$4.99 含 30 积分，US$9.99 含 70 积分，US$19.99 含 140 积分，均为每月。额外积分每个 US$0.175。每场会话最长可持续 60 minutes。

预测假设的会话时长为 20 minutes，用来估算使用量和每小时的经济性。每月额度按各规格的积分费率逐场会话消耗，而不是按小时计量。收益文档现在把这一假设与会话上限区分开，并链接到当前的规格说明。

Walter（walter.gg）可免费试用。本页说明已公布的订阅定价。