在Telegram上使用Polymarket？SkateChain如何将其无缝带入Ton生态

作者：Weilin，PANews

几天前，Skate的联合创始人、CEO Siddharth Lalwani在X上展示了项目的最新成果：在Telegram上参与预测市场Polymarket的押注，而无需切换链或者跨链转移资产。

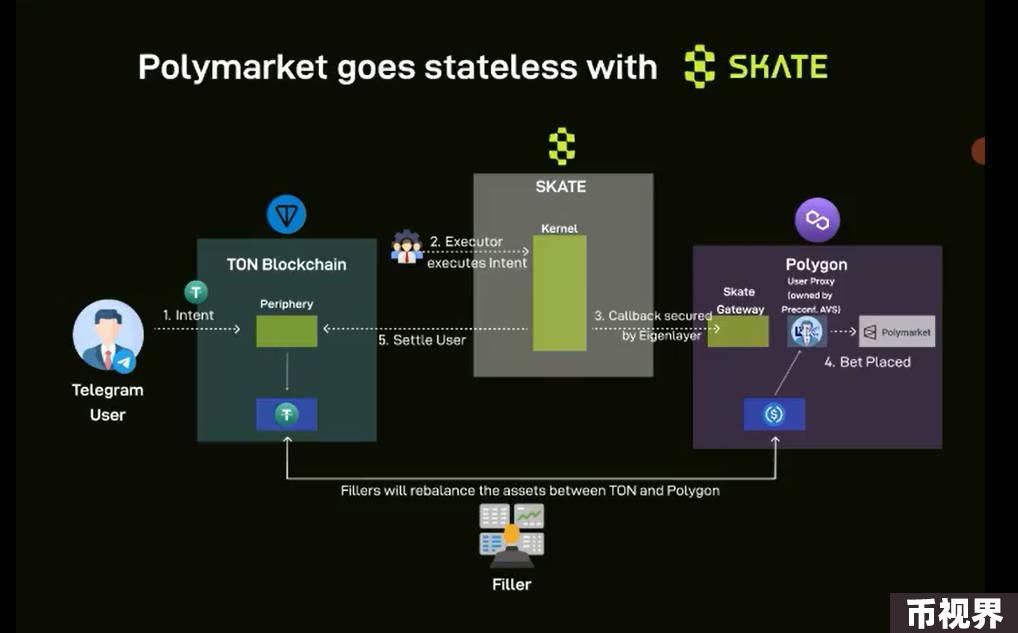
根据官方介绍，Skate是一个通用应用程序层，使应用程序能够通过一种状态在数千条链上运行。Skate监控每个网络的状态，当用户希望参与某种跨链操作时，用户签署“意图”。这些意图由执行者监控，执行者实时执行它们。例如，如果用户希望跨链转移10,000美元，执行者会从所有可用网络中寻求获取流动性，从而促进交易，实际上统一了流动性。

这依赖于中心辐射型架构，并利用EigenLayer来实现快速确定(fast finality)。这也意味着任何网络都可以轻松集成，因为Skate会监控它们的状态，然后将它们连接到整个网络。

本文，PANews对Siddharth Lalwani最新展示的连接Telegram和Polymarket的工作流程进行了图文编译，以便读者更好地理解这一创新。

Skate的联合创始人、CEO Siddharth Lalwani:

我们内部正在构建一个测试应用，用于在TON上启用Polymarket预测市场。用户可以直接在Telegram内，通过一个Telegram的迷你应用与Polymarket交互。他可以表达意图，比如他想下注10美元赌特朗普会赢得美国总统大选，整个流程都会为他自动执行。他不需要切换链，也不需要跨链桥接资产。



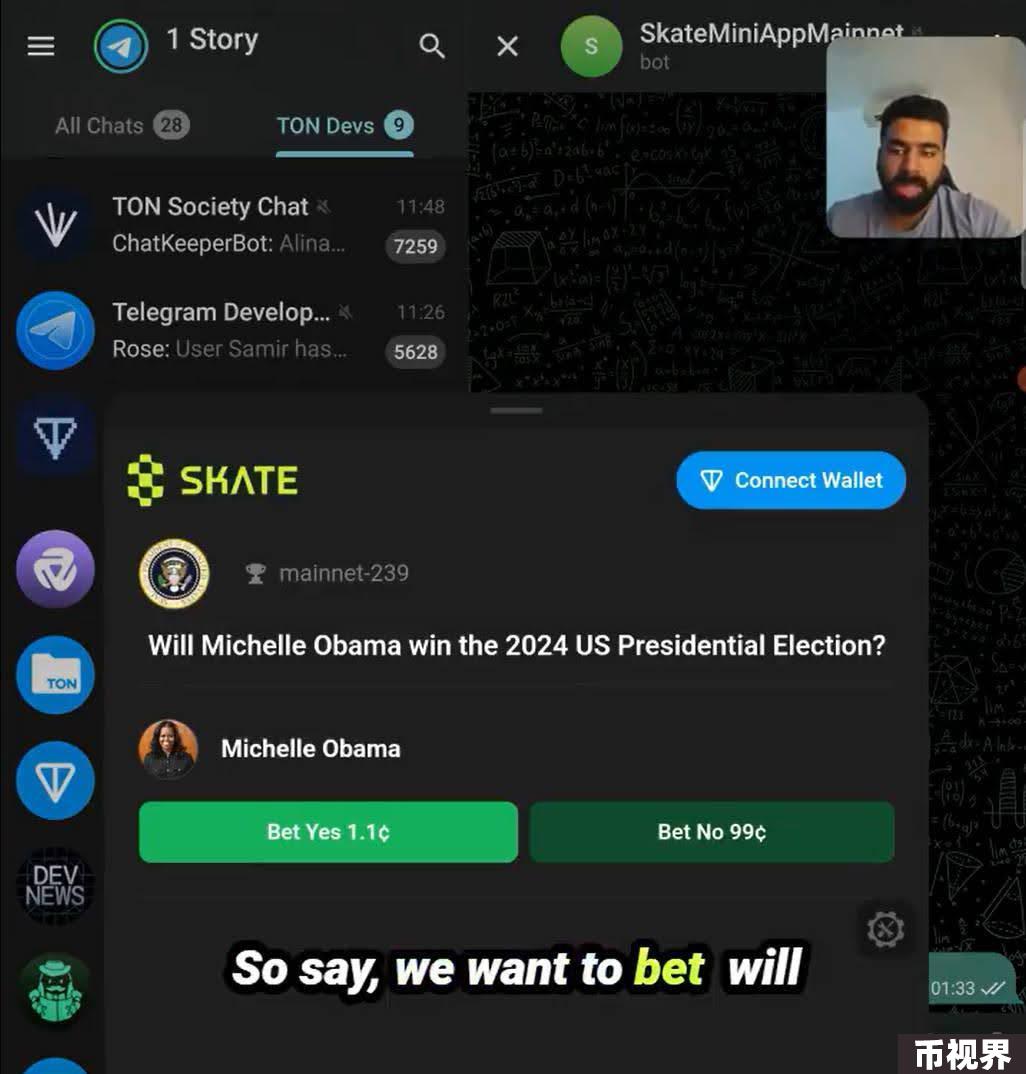
测试应用介绍：

对于用户来说，他仍然在Telegram内，通过迷你应用交互。他的资产仍在TON上，他只需要表达下注特朗普会赢得选举的意图。资产会进入一个中间阶段，并在TON区块链的外围合约(periphery contract)中。

总的来说，内核(kernel)代表核心应用逻辑和应用状态部分，而外围则代表用户交互逻辑部分。每个用户交互的链都会部署一个外围，但对于同一个应用，只有一个内核应用，这就是单一应用状态的原因。用户在TON上的Telegram迷你应用内表达意图，这个意图会被执行者拾取，并在Skate上部署的内核上执行。这个内核目前基本上存储了用户在Polygon上部署的代理合约、多签名钱包及其TON地址的映射，并启用从TON到Polygon的所有回调(callback)。

一旦需要回调，比如用户下注时，回调通过EigenLayer，利用从预确认AVS获得的高经济信任，回调会为用户创建代理合约，并在Polymarket上下注。

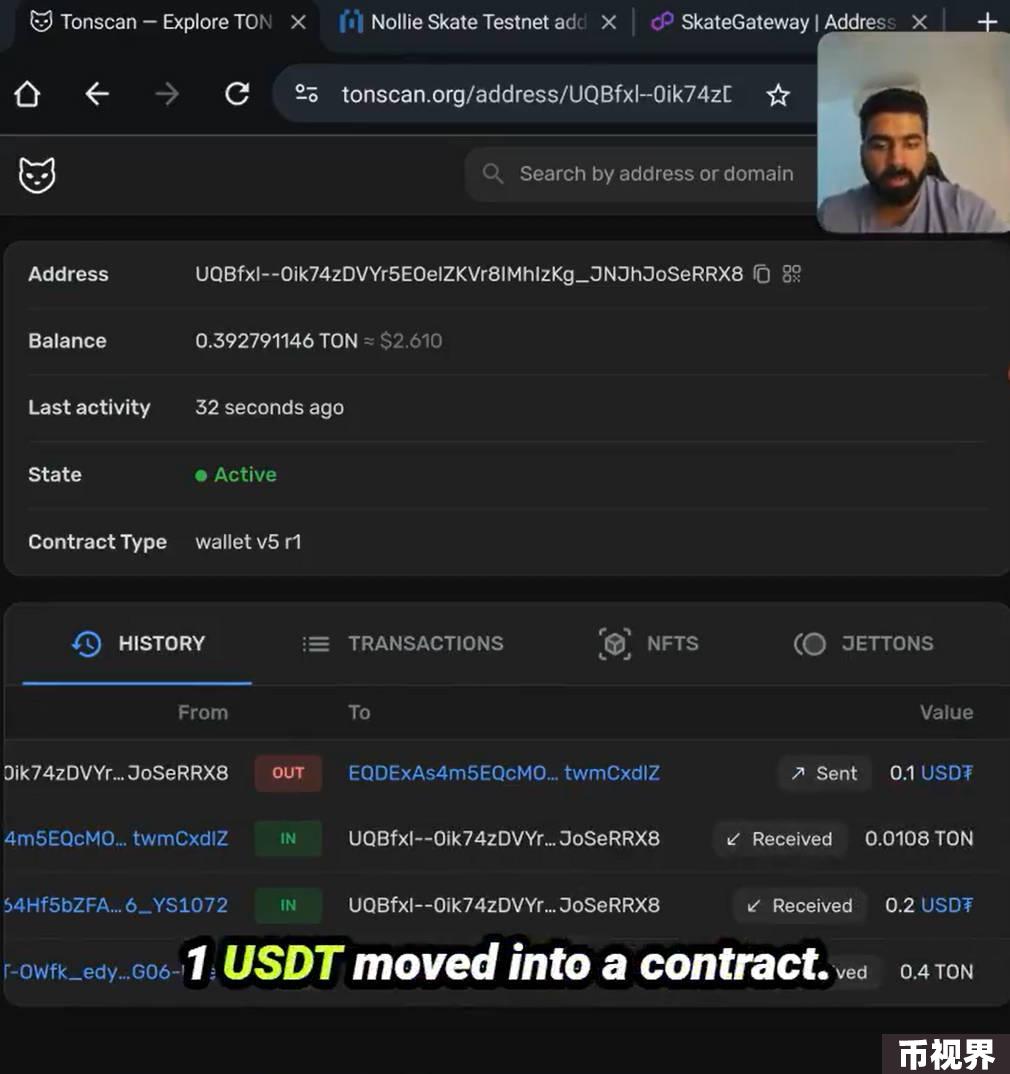
现在，假设我在Telegram内，通过我们开发的测试应用与之交互，下注米歇尔·奥巴马会赢得美国总统大选。我不认为她在参与竞争，不过我会快速演示。我们会将Tonkeeper钱包与应用连接，这是我创建的钱包，里面有一些USDT和用于gas的TON。假设我们下注0.1 USDT，赌米歇尔·奥巴马会赢得选举。这会生成一个下注意图。作为用户，这就是需要做的一切。

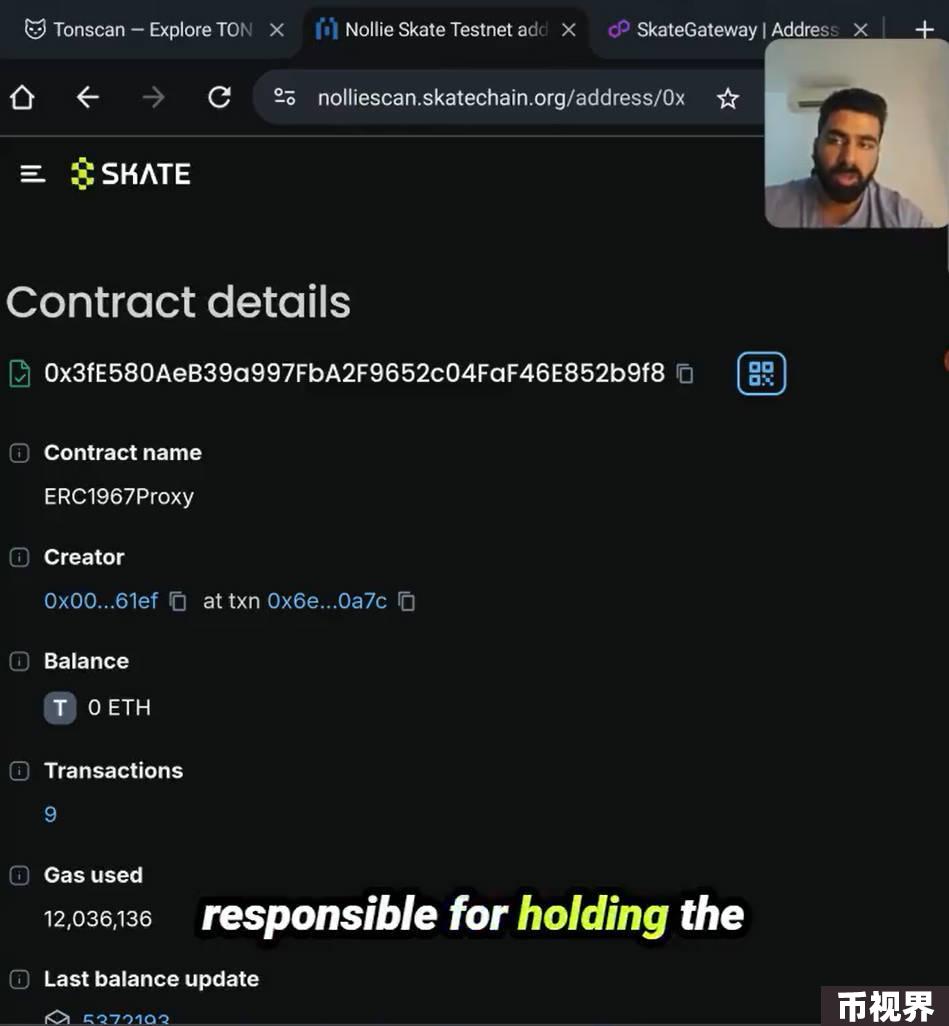


整个用户流程在后台的执行情况：

现在，通过区块链浏览器，我会快速演示后台实际发生了什么。我刚刚通过用户的钱包进行了交互，快速打开TON区块链浏览器查看交易。

这里显示了0.1 USDT转移到合约。这是中间阶段，现在为用户生成了在Polymarket上下注的意图。执行者会拾取这个意图，并在包含核心逻辑的合约上执行。

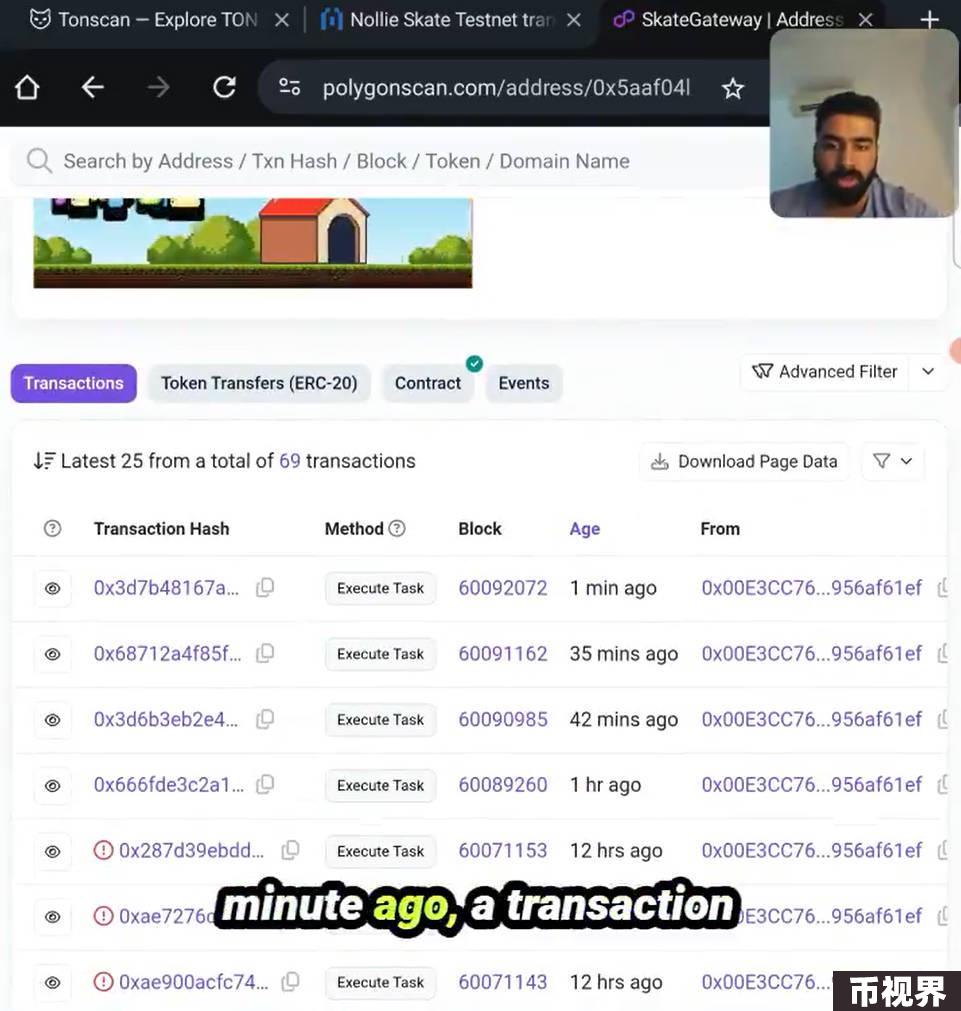




这是Skate链上的内核合约，这个内核合约负责保持与Polymarket交互的用户状态，通过Skate无状态应用模型完成。

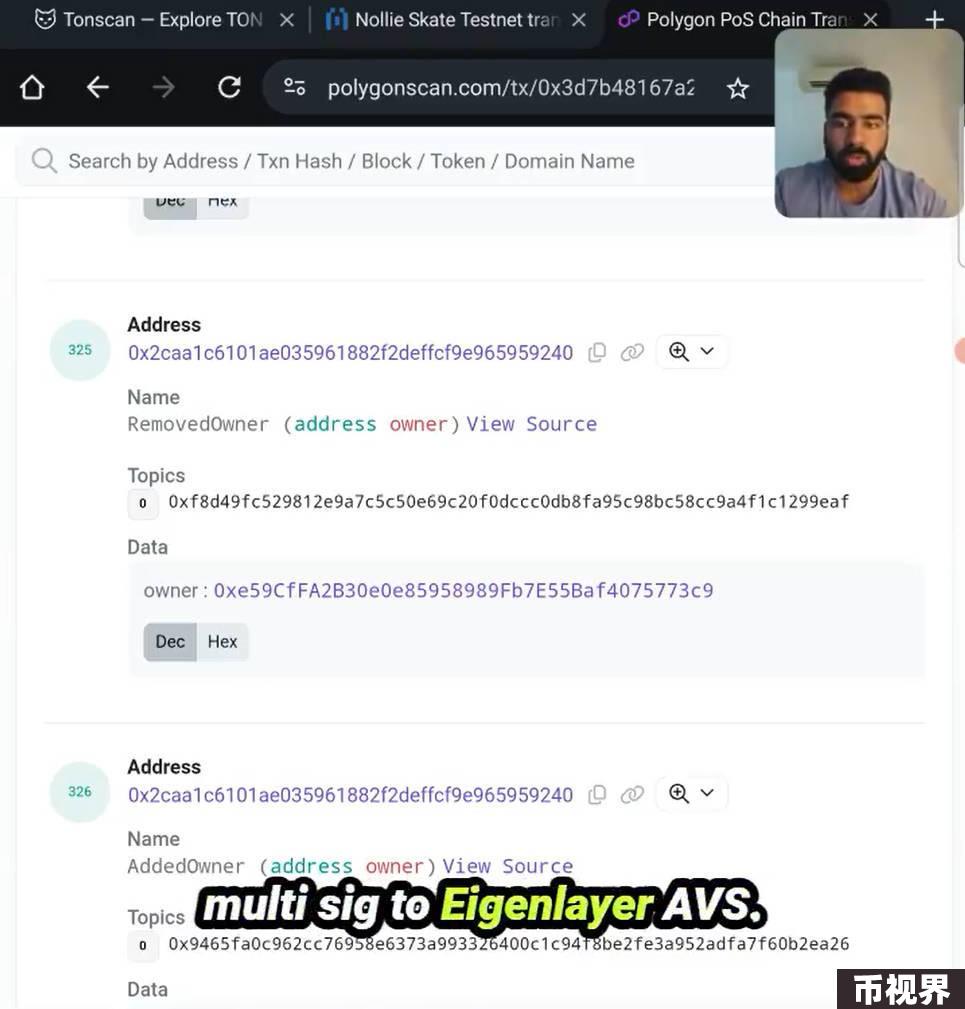
你可以看到，47秒前，内核部分的处理意图被调用。对于从Skate链到任何链的每个回调，都会创建任务并关联任务ID。AVS操作员会检查任务ID，并进行证明。一旦收到足够的证明，回调会在目标链上执行，这个案例中是Polygon。

因此，我们快速查看Polygon上的网关合约，所有来自EigenLayer AVS的交互，即预确认，将在这里部署的任何无状态应用外围上发生。快速刷新一下。你可以看到1分钟前，为用户执行了一笔交易。这次代表用户下注了0.1 USDT。我快速解释一下这里实际发生了什么。



首先，为用户创建了一个多签名钱包。我们自动生成了这个地址，并将其存储在SkateChain的映射中，这样代理就可以代表用户创建，并在映射中更新他们的TON地址。

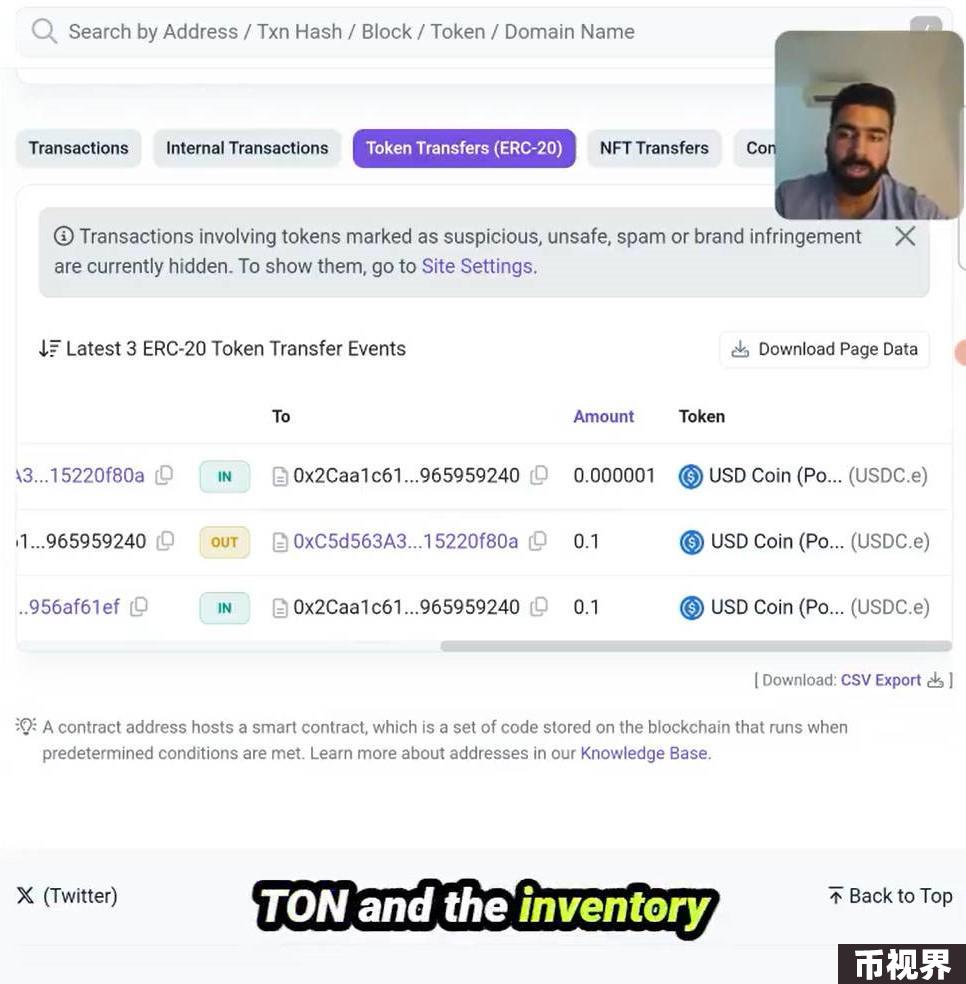
代理创建后，接下来执行的数据调用是你必须给Polymarket合约的不同批准。在这种情况下，是Polymarket上的CTF合约和一些其他互动必须进行的批准。批准完成后，接下来发生的是，我们首先从创建多签名的钱包的UA（User agent，指在进行网络通信时，代表用户端向服务器发送请求的一种标识）中移除所有权，然后将多签名的所有权转移给EigenLayer AVS。我认为这是我们设计中最优雅的部分。



现在，基本上，所有代表用户在不同链上交互的代理都由预确认AVS拥有。如果我们上线生产时有一个50亿美元的基准经济信任，这基本上决定了移动任何资产的成本。所以，如果你想攻击这个系统，经济成本就是这样。

因此，任何资产的移动都会在获得价值超过50亿美元的质押后发生。然后，Polymarket上的订单匹配就会完成。

我可以快速展示在多签名合约上的情况。这是为这个TON用户创建的多签名合约。首先，一个提供者为他提供了一些资产以便结算。提供者基本上可以在库存之间不断再平衡，以获得TON资产，最终他们会得到Polygon上的资产。你可以看到订单匹配已经执行，基本上代表用户在Polymarket上下了注。



之后，回调会在TON上发生，用户在那里结算。整个流程完全无缝，所有托管都由AVS负责，这为我们提供了非常高的经济信任，我会说使整个系统无信任。我只是想快速记录下来，并帮助你们了解这个无状态用户体验的样子。

现在，基本上，通过这种方式，我们可以在任何蓝筹EVM应用或非EVM链（如TON、Solana）上启用这些应用。更重要的是，我们可以启用这种新的异步应用设计模式，拥有单一应用状态。感觉就像你在所有链上运行同一个应用，但在用户层面，他不需要担心切换链。即使在应用开发者层面，他们也不需要担心。我在这个特定的链上，但现在有一个新链提供空投激励，我想把所有用户都迁移过去。现在你可以在所有链上运行你的应用，并获得所有生态系统的网络效应。

Siddharth Lalwani表示，“我们非常兴奋能够为大家提供这些应用，期待与大家的互动。我认为我们还有一些细节需要处理。我们希望听到你们的反馈，希望你们与任何有兴趣在这个新设计空间中构建的团队联系。我们非常高兴与大家分享这一切，期待社区的反馈。”