比特币上的资产发行：现有项目及各自方案指南

来源：字节元 CKB

我知道，一谈到这个问题，比特币纯粹主义者可能觉得：比特币就安静地做数字黄金不好吗？为什么一定要有代币呢？为什么非得有 USDT? 不过, 如果你特别在意资产安全的话，就不得不想，以太坊万一倒了呢？DeFi 谁能接住? 而且，代币方案与比特币协议是兼容的，并不会破坏原本的功能，如果不喜欢，可以不下载代币客户端，也不会受到很大的影响。

## 在比特币上发行代币：为何不可？

在比特币上发行代币，用来将现实世界中的资产交易转移到链上，这个想法在 2010 年左右就在比特币社区出现了。社区最初的讨论是设想把现实世界资产——比如：房产、股票、法币等资产，都搬到比特币上进行去中心化的交易。不过由于法律因素，房产和股票这类资产的搬运没那么容易。就算你将自己的房子的数字资产代币支付给了另一个人，政府可能不会承认，或者自动更换现实世界的房产证，还可能需要交各种税。况且在监管之下还不能随意在链上交易。

因此，更吸引人的方法是发行同法币挂钩的代币 ，即稳定币。稳定币与 NFT 不同，它们仍然是同质化的（fungible）代币，只不过与原本的比特币做了区分。当作为代币出现时，它们的价值由所代表的现实世界资产的价格决定，不再有原本的数字货币价格（如果数字货币的价格涨到比资产价格高太多，舍弃掉资产也不是不行）。这就是为什么通常比特币上的代币都会以聪（Satoshi）为单位。

将数字货币作为资产的代币，需要解决两个主要问题：

如何用比特币表示现实世界中的资产；

如何在比特币十分有限的脚本语言中设置复杂的交易规则和合约。

下面的内容着眼于以上两点，对目前现有的几大比特币资产发行方案做了概括，并从数据可用性、 资产载体、表现力、可扩展性等几个方面做了比较。

## 比特币上的第一个代币：染色币

最早在比特币上设计代币协议的人已不可考，想法可能产生于比特币论坛或社区里的讨论中。染色币（Colored Coin）项目是 Yoni Assia 在 2012 年发起的，当时他与 Vitalik Buterin、Lior Hakim、Meni Rosenfeld、Rotem Lev 一起写了《染色币白皮书》（Colored Coins whitepaper）[1], 项目在 2013 年开始运行。

染色币的工作原理是将一个聪标记成为一个特殊的钱币, 把资产的相关信息写到这个聪中——这个过程就叫染色。你可以将一个聪染成不同的颜色, 打上不同的标记（tag）, 不过同一种颜色下的硬币之间还是不能区分的，比如一堆染色成美元的聪，仍然是同质化的。比较早的协议使用的是 nSequence 字段，在交易的第一个 input UTXO 的 nSequence 中加入一个标记。不过 nSequence 存储上限只有 4 字节，所以后来的代币设计基本都换成了 OP\_RETURN 字段，能储存更多元数据。

染色币目前被提起主要还是因为它是比特币上的第一个代币项目。由于项目的发展其实并不理想，也没有得到大规模的应用，项目本身逐渐就被遗忘了。染色币在当时面临的问题就是比特币的功能还不能支持这个比较超前的想法，这个想法要落地，要高效稳定运行是很难的。这可能也是为什么 Vitalik 在染色币项目之后走向了比特币的反面，对智能合约那么执着。

由于染色币是以聪的形式存在的，它的验证就和验证一个 UTXO 的有效性一样，都需要下载整个链。这个问题在后面会以客户端验证（client-side validation）的方式解决。

## 用 OP\_RETURN 发代币：Counterparty & Omni Layer

和染色币不一样，Counterparty[2] 和 Omni Layer[3]（ USDT 背后的协议） 并不直接在聪上染色，而是在交易中设置一个数值为 0 的 UTXO，在这个 UTXO 的 OP\_RETURN 中存放元数据。OP\_RETURN 可存放 80 个字节，标记了 OP\_RETURN 的 UTXO 不能被花费，真正的代币是 OP\_RETURN 中记录的 i-th output。这个 output 的数值通常是 0.00000546 BTC——系统允许发送的最低值，而且由于代币的价值并不与 BTC 挂钩，并没有必要发比 0.00000546 BTC 多的币值。

这些项目的验证都需要在链上进行，元数据储存在链上。

Omni Layer 在很长一段时间都是以太坊链上的玩家，直到最近才回到比特币生态，准备发行 BTC-USDT。Counterparty 质押了一部分比特币，有自己的代币 XCP。从 Twitter[4] 上看最近应该是在做 NFT。

进一步了解 OP\_RETURN，可参考：

An analysis of Bitcoin OP RETURN metadata[5]

手动构建 OP\_RETURN 发送 USDT[6]

## 用侧链锚定比特币：Rootstock & Liquid Network

Rootstock[7] 和 Liquid Network[8] 这两个项目大约出现在 2017 年左右，都是侧链方案——用双向锚定（Two-way peg）的方式把比特币置换到侧链，并在 EVM 兼容的侧链上使用各种 DeFi 和 dApps。他们有类似 WBTC[9] 的代币 （RSK 有 RBTC，Liquid 有 L-BTC)，主要面向的是想用 BTC 在以太坊生态上 build 的人。

在 Rootstock 上发行代币，方法与在以太坊上发行是一样的，或者可以说 Rootstock 这个侧链除了挖矿是与比特币链一起，其他的功能都是为适配以太坊生态做的，比如智能合约代码也是用 Solidity 写。所以这里的代币都是在 RBTC 基础上发行的，并不直接和 BTC 有联系。

由于本文主要关注公链，而 Liquid Network 是一个联盟链，这里不深入讨论。

进一步了解 RSK，参考：

RSK: A Bitcoin sidechain with stateful smart-contracts (RSK paper)[10]

RSK money[11]

FAQs[12]

前面提到的这些项目, 有一些消失了（比如染色币），有一些打着比特币的幌子卖的是以太坊的生态。这主要是因为以太坊在拥抱资本之后，DeFi 和 dApps 占据了绝对的市场优势，所以不和它玩的 DeFi 项目想要获得优势就比较困难。以太坊上的代币是通过合约来发行和交易的，遵循 ERC-20 等标准。比特币生态在最近两年也开始解锁合约功能，如 BitVM，也有代币标准 BRC-20 出现。

## 在比特币上实现智能合约：RGB

诞生于 2016 年的 RGB（Really Good for Bitcoin）[13] 最初被设计为染色币的竞争对手。但面对类似的挑战，它转向在比特币上启用智能合约。尽管它主要关注的是运行智能合约，而非发型代币，但由于它们的虚拟机 AluVM 的限制，截至 2024 年，完整的合约功能仍然有限。

RGB 的思路是把能拿到链下的数据和智能合约代码都放在比特币之外进行，通过 Merkle root 来提供 交易验证和代币发行的承诺（commitment），比特币链只做交易承诺的验证和最终性，证明没有出现双花。

RGB 值得一提的地方是同时使用了客户端验证和一次性密封条的技术，这样它并不在 UTXO 上做标记来表示代币。这两个概念最早是由 Peter Todd 在 2013 年[14] 提出的，Giacomo Zucco 和 Maxim Orlovsky 在这个基础上设计了 RGB 协议。

客户端验证（Client-side validation） 让交易使用的数据和代码都保存在链下，不会公开广播，有些数据可能只会在交易双方之间私下交换，其他与交易不相关的人可能毫不知情。链下状态的借助比特币维护，区块链是作为时间戳发挥作用的，可以证明状态的先后次序。

而一次性封条（single-use seal）——它也是客户端验证最常出现的样子——是数字版的一次性密封条。它借助每个 UTXO 只能被花费一次的性质，把链下状态的信息写到一个 UTXO 中。这样如果某个时刻这个 UTXO 被花掉了，我们就知道状态被更新了，更新之后的状态信息写到新生成的 UTXO 中。这个链下状态信息可以是 USDT 代币的所有权，也可以是某个合约中有多少代币。

比如 Alice 想把一个 USDT 转给 Bob，这个 USDT 并不是存在比特币链上，它的信息是在链下维护的，但是它会和一个由 Alice 控制的 UTXO 有联系。它的信息保存于生成这个 UTXO 的那笔交易中 数值为零的 UTXO 的 OP\_RETURN 字段中。这样，只有 Alice 能花掉这个 USDT，而且 Bob 可以通过链上的交易追踪到这个 USDT 在过往交易中曾被保存在哪些 UTXO 中，这些 UTXO 是不是有效的，以及交易是不是合法的。这样，当 Alice 发起交易，把这个 USDT 的 承诺信息转移到一个由 Bob 控制的 UTXO 时，Bob 就可以确定他获得了这个 USDT。

RGB 也可以在闪电网络上运行，因为它的状态是链下的，只需要把承诺放到链上或者闪电网络上。在 Taproot 升级之后，RGB 可以把承诺嵌入到一个 Taproot 交易中，这可以让 RGB 以更灵活的方式将承诺 嵌入到比特币链上。

进一步了解 RGB，参考：RGB Blueprint[15]

## 只支持代币不支持智能合约：Taproot 资产

Taproot Asset 是 Lightning Network Daemon (LND)[16] 团队开发的项目。它的原理和 RGB 类似，但并不支持复杂的智能合约，只支持代币（参考这里对 Taproot 词条[17] 的解释）。

进一步了解 Client-Side Validation、RGB 和 Taproot，参考：

Client-side validation[18]

Off-Chain Transactions: The Evolution of Bitcoin Asset Protocols[19]

Counterparty vs RGB vs TARO[20]

## 让每个聪都与众不同：Ordinals & Inscriptions

Casey Rodarmor 在 2023 年初发布了 Ordinal Protocol[21]。这个项目最初是从这样一个想法而来：如何给聪编号，让每一个聪都有一个独一无二的序列号从而被排序。这个想法和染色币是同时期的，只是在去年才被再次提出。而且由于 SegWit 和 Taproot 功能的加入，它的实现变得不那么难了。Ordinal 让每一个聪都彼此不同，这就使得 NFT 可以直接在比特币链上发行。

Inscriptions[22] 就是一个这样的 NFT 项目。NFT 的数据保存在交易的 witness 数据中，而不是之前项目使用的 OP\_RETURN 字段，这样可以存下大小为 4MB 以内的元数据。与以太坊上 NFT 不同，Inscription 是链上存储，包括元数据和图片。

进一步了解 Ordinals，参考：

Ordinals: A common ground for Ethereum and Bitcoin maximalists?[23]

The Ultimate Guide to Bitcoin Ordinals and Inscriptions[24]

## 双向绑定任一 UTXO 链：RGB++ 同构绑定

RGB++[25] 最初是作为 BTC 与 CKB（Nervos Network[26] 的基础）之间的同构绑定协议（isomorphic binding protocol）出现，而现在它的适用范围很广，不是只局限于 CKB 和 BTC 之间，只要是两个 UTXO 链理论上都能用这个协议绑定在一起。

RGB++ 将 RGB 的 Client-Side Validation 和 Single-Use-Seals 思路做了近一步发挥。如前所述，RGB 协议最大的问题就是数据由用户自己保存在本地。如果用户不小心把数据弄丢了，是没有备份，也找不回来的。而且，由于用户只保存和自己的代币相关的数据，其他数据想要验证就比较难。同构绑定层的方案就是不仅仅把代币绑定到比特币 UTXO 的 OP\_RETURN 字段中，也把相应的比特币交易信息绑定到 CKB 链上的交易里（通过在 CKB Cell[27] 的 Lock Script 里，使用一个特殊的 IB-lock-script 而实现）。当判断 CKB 链上的交易是否合法时，Lock Script 会用 CKB 上 BTC light client 的数据，看对应的 UTXO 有没有被花费，以及被花掉之后新生成的 UTXO 是不是绑定了目前这笔的代币交易信息（作为不含签名的部份信息）。

RGB++ 值得关注的特点：

通过双向绑定解决数据可用性问题：CKB Cell 承诺绑定在 UTXO 的 OP\_RETURN 字段；UTXO 信息绑定在 CKB 交易的 Output Cell。

与闪电网络和 Fiber Network（基于 CKB 的闪电网络）兼容

支持多资产

可以和任何 UTXO 链绑定

进一步了解 RGB++，参考：

RGB++ Protocol Light Paper[28]

The Ultimate Guide to RGB, RGB++, and Client-Side Validation[29]

为了更清楚地了解各项目的优势和局限，我们将以上项目放入下面的表格中比较。其中需要重点关注的指标有：

数据可用性（Data availability）：同构链（isomorphic-chain）和侧链相差无几，而链下的数据可用性要弱于其他方案。此项从强到弱的排序为：链上 ≥ 同构链 ≥ 侧链 > 链下；

资产载体（Asset carrier）：直接同 BTC 关联的代币方案要优于非直接关联的方案；

同质性（Fungibility）：这里指的是项目的原生代币是否可相互置换，并不是说项目不支持发行 NFT，后者可以通过增加额外协议来实现；

表现力（Expressiveness）：指处理复杂智能合约的能力。

文中提到的部分链接：[1]https://docs.google.com/document/d/1AnkP\_cVZTCMLIzw4DvsW6M8Q2JC0lIzrTLuoWu2z1BE/edit?pli=1&tab=t.0#heading=h.pr8n14cpqri5

[2] https://www.counterparty.io/

[3] https://www.omnilayer.org/

[4] https://x.com/counterpartyxcp

[5] https://arxiv.org/pdf/1702.01024

[6] https://juejin.cn/post/6844903769327534088

[7] https://rootstock.io/

[8] https://liquid.net/

[9] https://www.wbtc.network/

[10] https://eprint.iacr.org/2022/684.pdf

[11] https://wiki.moneyonchain.com/getting-started/what-do-i-need-to-know-first

[12] https://dev.rootstock.io/resources/faqs/

[13] https://rgb.tech/

[14] https://petertodd.org/2013/disentangling-crypto-coin-mining

[15] https://rgb-org.github.io/

[16] https://docs.lightning.engineering/

[17] https://bitcoinops.org/en/topics/client-side-validation/

[18] https://bitcoinops.org/en/topics/client-side-validation/

[19] https://mirror.xyz/0x5CCF44ACd0D19a97ad5aF0da492AC0388469DfE9/h7XChxETK-cBfGdc0sTq\_cCuWeo13-sp1j-g0ZYoYdc

[20] https://mandelduck.medium.com/counterparty-vs-rgb-vs-taro-8cd707d544f7

[21] https://docs.ordinals.com/

[22] https://ordinals.com/inscriptions

[23] https://blog.kraken.com/crypto-education/ordinals-a-common-ground-for-ethereum-and-bitcoin-maximalists

[24] https://www.nervos.org/knowledge-base/guide\_to\_inscriptions

[25] https://www.rgbppfans.com/

[26] https://www.nervos.org/

[27] https://docs.nervos.org/docs/tech-explanation/cell

[28] https://github.com/utxostack/RGBPlusPlus-design/blob/main/docs/light-paper-en.md

[29] https://www.nervos.org/knowledge-base/ultimate\_guide\_to\_rgb\_rgbpp\_and\_client\_side\_validation