

如何为btc链上资产编写索引器

以染色币协议atomicals为例

From: Spike X:@isyiming 2025/07/09

井字棋游戏的启发

数据在纸上更新：存储介质

规则在玩家脑海：游戏规则



最具价值的资产btc和在btc上发行的资产

- Btc网络的安全性
 - 透明性，所有数据公开可验证
 - 分布式存储，没有单点故障
 - Pow共识，不可篡改
 - 更加牢不可破的共识，根植于人们脑海
- Btc链上数据
 - 存储于btc链上，天然获得以上特性：
透明性，安全性，不可篡改
 - 如何形成共识？井字棋的游戏规则->资产的有效性认证->索引器



最具价值的资产btc和在btc上发行的资产

协议名称	BCR-20	BITCOIN RUNES	ATOMICALS
作者	domo	Casey	Arthur
基础机制	witness	OP_RETURN	witness+utxo

- 以上协议都将指令写在UTXO中，数据上链了；但是BTC协议本身不支持这些指令，也不关心这些指令的含义
 - BRC-20在纸币上做如下标记：Instruction 1. mint ordi； Instruction 2. transfer to Spike 1000 ordi
 - Atomics/Runes在纸币上做标记 Instruction1. mint atom；并将纸币送给了Spike（不需要新的指令）
- 需要外部机制解释并规范这些指令：索引器

BTC上的染色币协议 Atomicals

- 如何染色
 - 上一笔UTXO转账时，在witness字段写入指令：mint atom
 - 下一笔交易的UTXO即被染色
 - Btc的UTXO的转移就代表atomicals资产的转移（染色币的魅力所在）
- Witness字段承担的功能
 - deploy / mint / split / 其他指令mod dat等

上一笔交易的UTXO

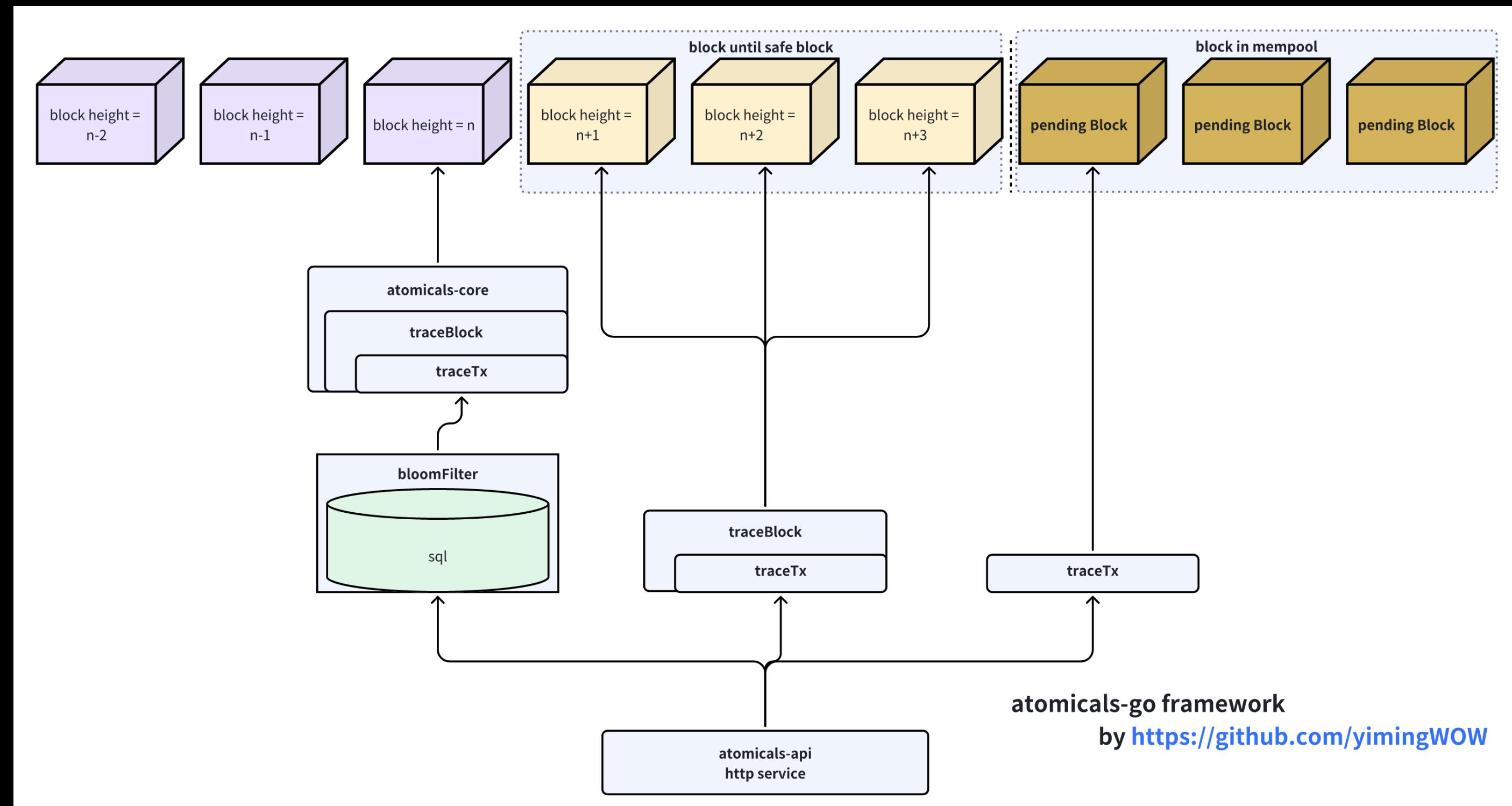
下一笔交易的UTXO

输入与输出

Witness	Inscription	ScriptPubKey (ASM)	ScriptPubKey (HEX)	类型
<pre>bc1p478w38t7reqdhxup94... 7sg44vh</pre>	US\$102.08	<pre>OP_PUSHNUM_1 OP_PUSHBYTES_32 7bcd9752cbeaf9de821e23f112597f 22bdf0c0cae23543cccb3f27e6016cf38</pre>	<pre>51207bcd9752cbeaf9de821e23f112597f22bdf0c0cae2 3543ccb3f27e6016cf38</pre>	V1_P2TR
<pre>bc1p009ajafvh6hem6ppugl3zfvh...uqprk40v</pre>	US\$0.36			
<pre>c0bb8fe8a0413bd58d58fe607c9f260a2d106825dfafda 32b928ca8e9bf94c290bac0063036f7264010118746578 742f706c61696e3b636861727365743d7574662d38003d 7b2270223a226272632d3230222c226f70223a226d696e 74222c227469636b223a226b6f686f222c22616d74223a 22333432383537313432383537227d68</pre>	US\$2.62			

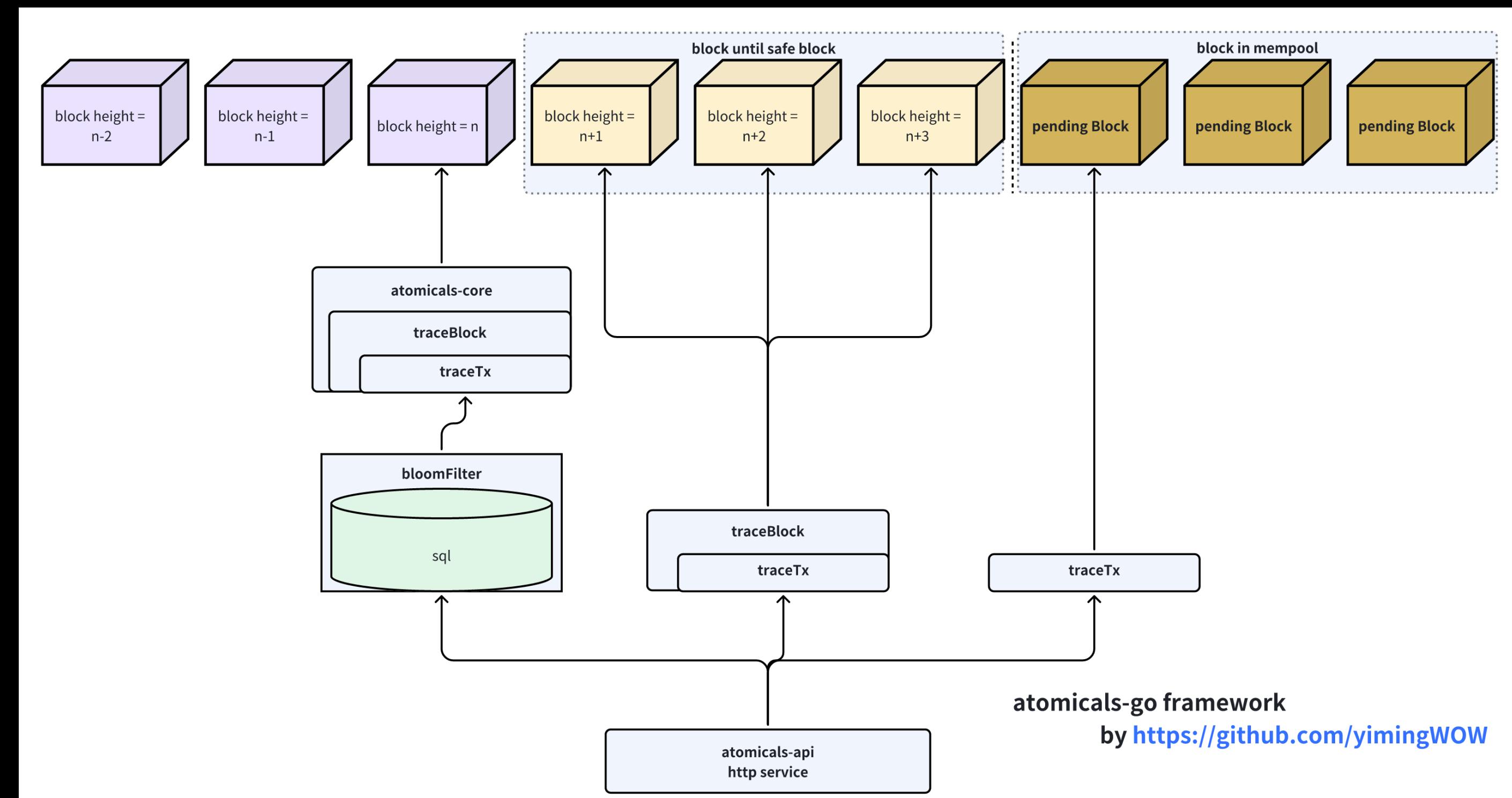
Atomics 索引器核心内容

- 区块同步 →
- 交易解析 →
 - 解析链上指令
- 业务处理 →
 - 协议的核心逻辑，判断有效交易，剔除无效交易
 - 能否并发？
- 数据存储
 - 每捕获一笔有效的Atomics交易，就立即更新存储
 - 并记录当前处理blockHeight txIndex,保证断点重连



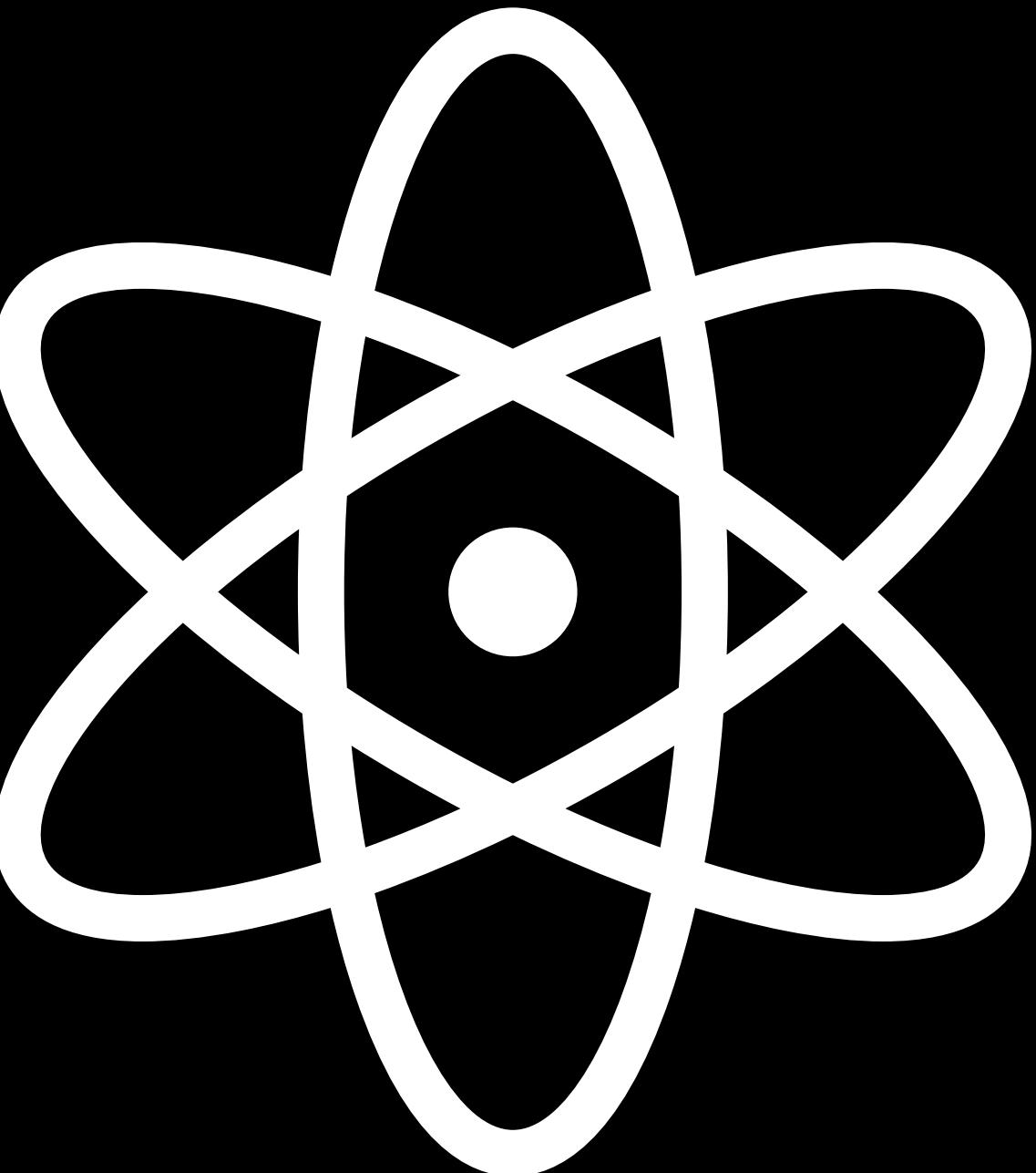
Atomicals 索引器经验之谈

- 从atomicals开始部署的高度开始同步数据
 - 确实有人会从中本聪创世区块开始检索:(
- 并发读取btc交易，维护一个合理的待处理队列
 - 保证从内存中读取交易
- 安全区高度内的交易才落库
 - DB中的数据+安全高度以上的交易 即可得到atomicals资产的最新状态
 - 延迟落库避免分叉影响问题
- 捕获一笔有效的Atomicals交易后，更新存储
 - 并记录当前处理blockHeight txIndex,保证断点重连
- 新的atomicals资产有效性判定以来数据库数据
 - 使用布隆过滤器过滤大量无效utxo
 - Db也要维持一个合理大小的缓存，否则索引器同步会很慢



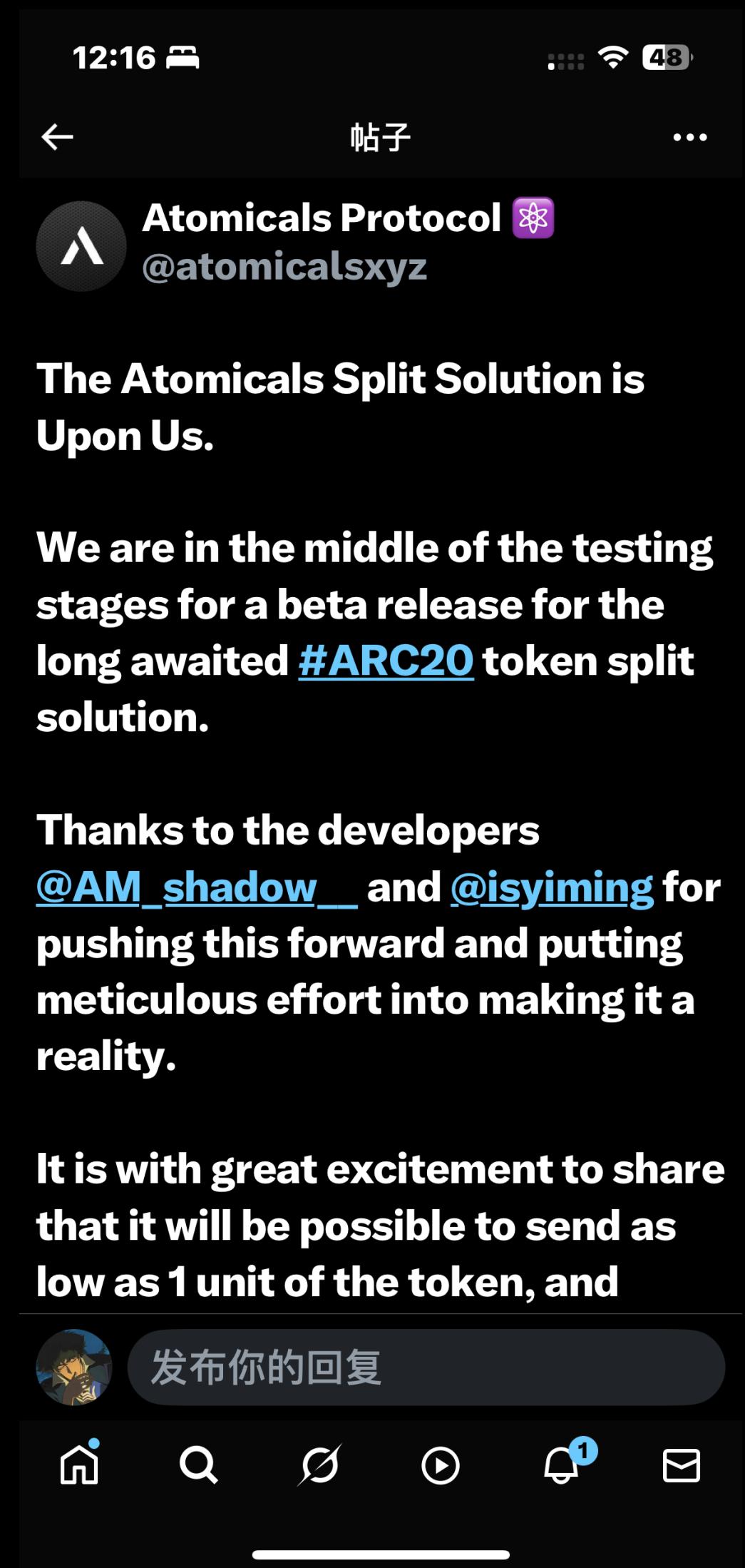
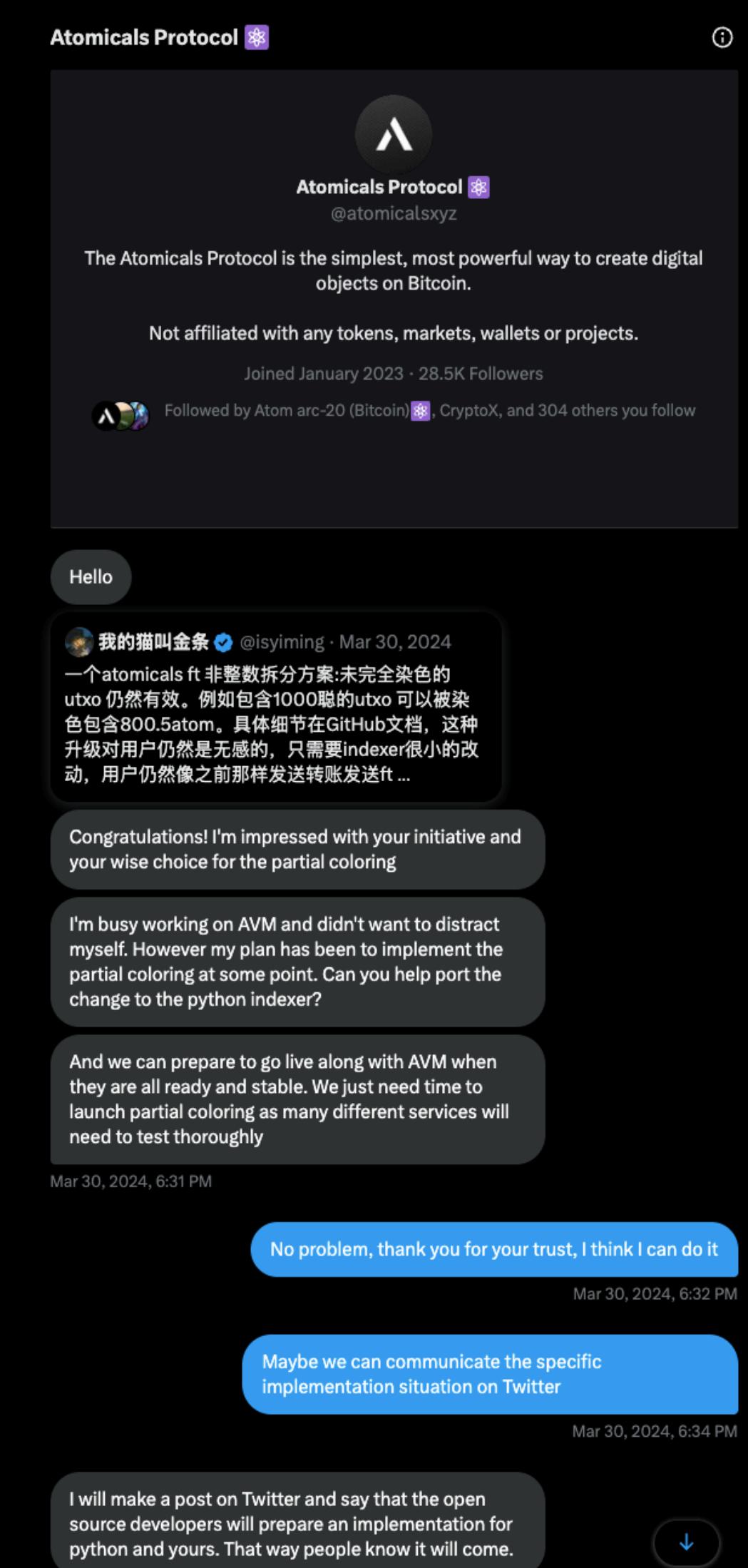
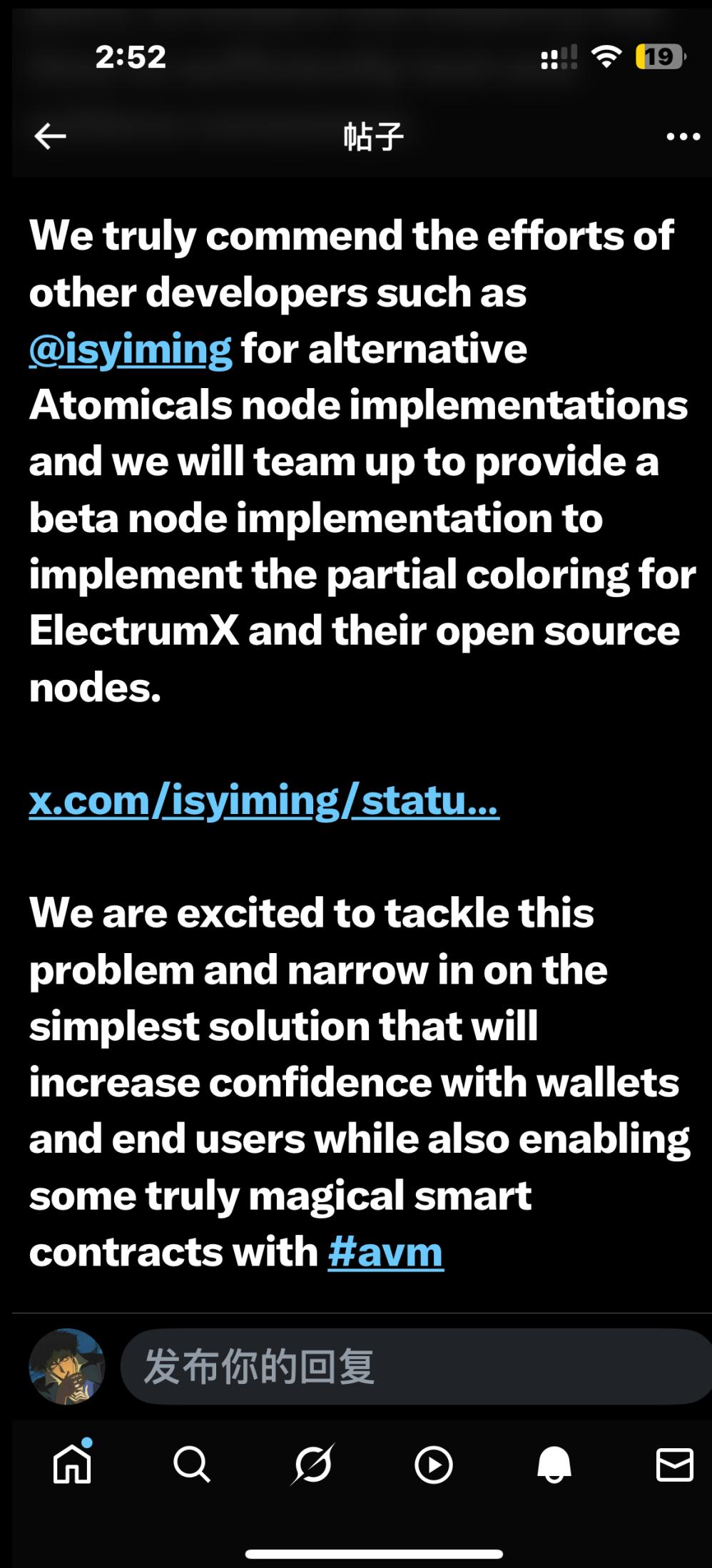
如果让你设计一个新的协议

- 顶层设计，全面考虑，留下缺省字段和待使用指令
 - 为将来拓展和更新留有余地
- 简洁的指令，不要留下历史包袱
- 新的交易不要过度依赖历史交易
 - 严重拖累索引器性能
- 尽量利用上btc的原生特性
- 染色币仍然需要索引器，但是transfer时不需要额外指令已经让人非常着迷了



哦我和Atomicals

- atomicals-go go版本的索引器
 - <https://github.com/atomicals-community/atomicals-go>
 - Arthur 发推介绍
- 针对ft拆分，提案atomicals 部份染色方案



Thanks a lot

BTC 生态还有新的可能性吗？