HTX VENTURES:
EXPLORING THE BTCFI RABBIT HOLE FROM THE PERSPECTIVE OF BITCOIN'S PROGRAMMABILITY



Abstract



Singapore / September 12, 2024 – Starting with the feasibility and evolutionary path of Bitcoin programming, this paper systematically explores Bitcoin's potential and challenges in the field of Bitcoin decentralized finance (BTCFI). Bitcoin adopts the UTXO model in its architecture and has established a contract system with verification at its core through its unique scripting language and opcodes. Compared with Ethereum's smart contracts, Bitcoin contracts are characterized as "stateless" and "non-computational", which limits their functionality but makes them more secure and decentralized.

With the implementation of the Taproot upgrade, Bitcoin's contract capabilities have been remarkably enhanced. The introduction of Taproot, especially the application of MAST and Schnorr signatures, enables Bitcoin to support more complex contract logic and significantly improves privacy and transaction efficiency. These technological innovations pave the way for BTCFI's further development, empowering Bitcoin to explore more financial application scenarios while maintaining its original decentralized advantages.

On this basis, this article delves into how Bitcoin programming can support various BTCFI applications. Through interpretations of mechanisms such as multisig, timelocks, and hashlocks, as well as discussions on the application of tools like DLC, PSBT, and MuSig2, the article demonstrates the possibility for Bitcoin to achieve decentralized liquidation and complex financial contracts under a trustless premise. This decentralized financial system, native to the Bitcoin network, not only overcomes the centralization risks of cross-chain bridging models like WBTC but also provides Bitcoin holders with a more solid trust foundation.

The article emphasizes at the end that Bitcoin's development in the decentralized finance field is not just a technological advancement but a profound change in its ecosystem structure. As new applications such as the Babylon staking protocol spring up and native scaling methods like Fractal Bitcoin come online, BTCFI's market potential is gradually becoming evident. In the future, with the rise in Bitcoin's price, BTCFI will further attract mainstream users, fostering a new financial ecosystem centered around Bitcoin. The formation of this ecosystem will propel Bitcoin further beyond the "digital gold" narrative, establishing it as an indispensable decentralized financial infrastructure in the global economic system.



Foreword

Since the launch of the Ordinals protocol in December 2022, the market has welcomed dozens of asset issuance protocols such as BRC-20, Atomicals, Pipe, and Runes, as well as hundreds of Bitcoin Layer 2 networks. At the same time, the community has been actively discussing the feasibility of Bitcoin decentralized finance (BTCFI).

In the previous crypto cycle, WBTC emerged to attract Bitcoin holders to participate in DeFi. WBTC locks Bitcoin through centralized custodial institutions and mints it into WBTC for use in Ethereum's DeFi protocols. WBTC's target users are Bitcoin holders who are willing to bear the risks of centralized cross-chain bridges to participate in Bitcoin DeFi.

As a typical example of bridging Bitcoin to the EVM ecosystem, WBTC represents one pathway to realizing BTCFI. The EVM Bitcoin Layer 2 networks and the DeFi projects in their ecosystems that have appeared in this cycle have also continued this model. Due to this model, WBTC has achieved a market value of over $9 billion in the Ethereum ecosystem, but it accounts for less than 1% of Bitcoin's total market value, which reflects the model's limitations.

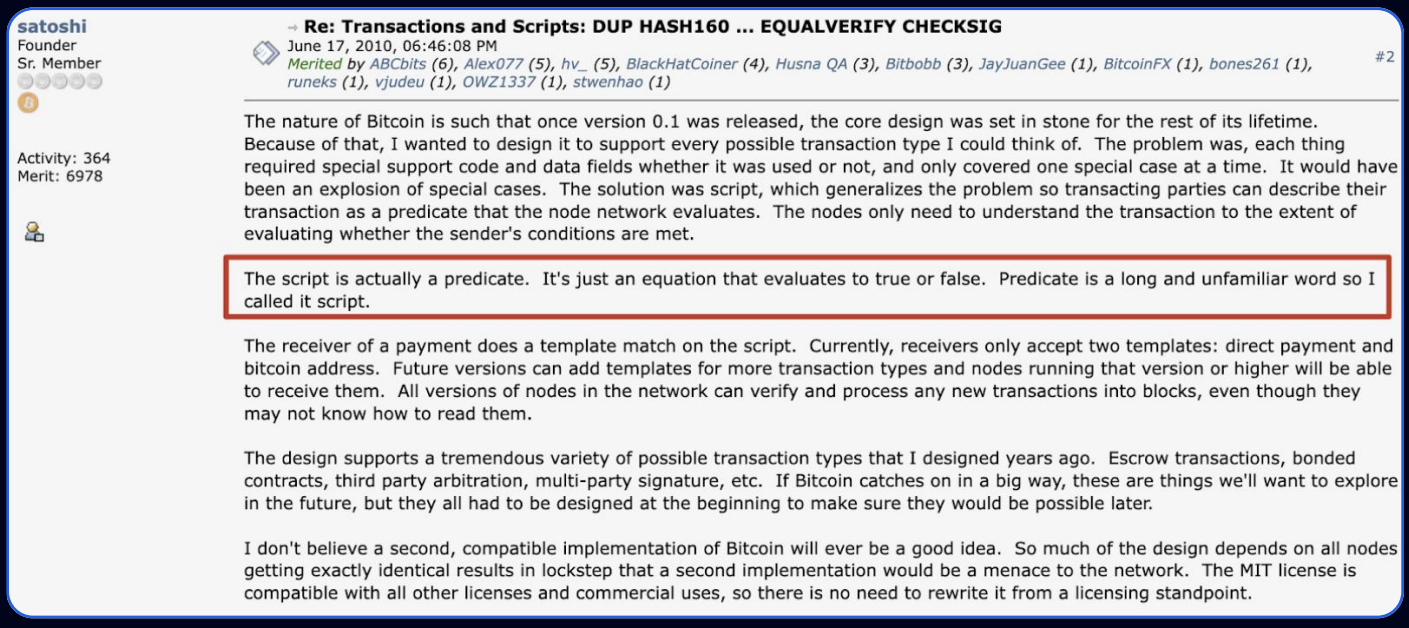
In comparison, if Bitcoin holders could directly participate in BTCFI using Bitcoin without the need for cross-chain minting, and meanwhile decentralized custody of funds could be ensured, it would attract more Bitcoin users and create a broader market. This entails achieving Bitcoin programming within the UTXO structure. Just as grasping Solidity is the key to entering Ethereum DeFi, mastering Bitcoin programming is an essential skill for entering the BTCFI market.

Unlike Ethereum contracts, Bitcoin contracts lack computational capabilities and are more like verification programs connected by signatures. Despite the initially limited application scenarios, Bitcoin programming has showcased increasingly evident potential with the ongoing upgrading of the Bitcoin network and innovations of the OG community. Many of its research results have already been transformed into BTCFI products set to go live.

This article will explore in depth the development path of BTCFI from the perspective of Bitcoin's programmability, clarify the history and logical framework of Bitcoin programming, and help readers understand the current real-world BTCFI implementation scenarios and how these scenarios will impact Bitcoin holders and the entire Bitcoin ecosystem.

Foundations of Bitcoin Contracts

Satoshi Nakamoto's Thoughts: UTXO, Scripting Language, and Opcodes



satoshi
Founder
Sr. Member
Activity: 364
Merit: 6978

Re: Transactions and Scripts: DUP HASH160 ... EQUALVERIFY CHECKSIG
June 17, 2010, 06:46:08 PM
Merited by ABCbits (6), Alex077 (5), hv_ (5), BlackHatCoiner (4), Husna QA (3), Bitbobb (3), JayJuanGee (1), BitcoinFX (1), bones261 (1), runeks (1), vjudeu (1), OWZ1337 (1), stwenhao (1)
#2

The nature of Bitcoin is such that once version 0.1 was released, the core design was set in stone for the rest of its lifetime. Because of that, I wanted to design it to support every possible transaction type I could think of. The problem was, each thing required special support code and data fields whether it was used or not, and only covered one special case at a time. It would have been an explosion of special cases. The solution was script, which generalizes the problem so transacting parties can describe their transaction as a predicate that the node network evaluates. The nodes only need to understand the transaction to the extent of evaluating whether the sender's conditions are met.

The script is actually a predicate. It's just an equation that evaluates to true or false. Predicate is a long and unfamiliar word so I called it script.

The receiver of a payment does a template match on the script. Currently, receivers only accept two templates: direct payment and bitcoin address. Future versions can add templates for more transaction types and nodes running that version or higher will be able to receive them. All versions of nodes in the network can verify and process any new transactions into blocks, even though they may not know how to read them.

The design supports a tremendous variety of possible transaction types that I designed years ago. Escrow transactions, bonded contracts, third party arbitration, multi-party signature, etc. If Bitcoin catches on in a big way, these are things we'll want to explore in the future, but they all had to be designed at the beginning to make sure they would be possible later.

I don't believe a second, compatible implementation of Bitcoin will ever be a good idea. So much of the design depends on all nodes getting exactly identical results in lockstep that a second implementation would be a menace to the network. The MIT license is compatible with all other licenses and commercial uses, so there is no need to rewrite it from a licensing standpoint.

Source: *https://bitcointalk.org/index.php?topic=195.msg1611#msg1611*

In 2010, Satoshi Nakamoto wrote on the Bitcoin Talk forum:

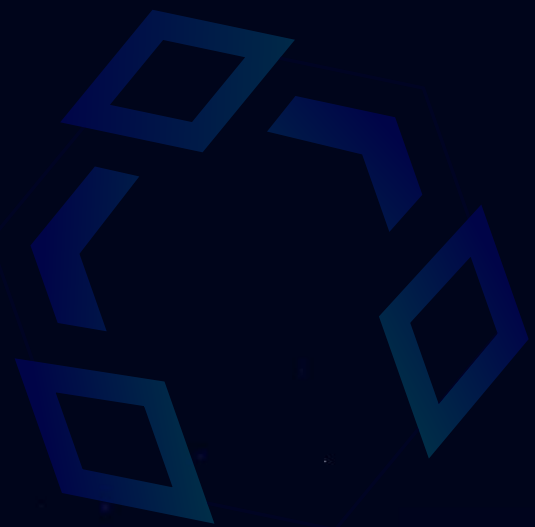
> *"The nature of Bitcoin is such that once version 0.1 was released, the core design was set in stone for the rest of its lifetime. Because of that, I wanted to design it to support every possible transaction type I could think of. The problem was, each thing required special support code and data fields whether it was used or not, and only covered one special case at a time. It would have been an explosion of special cases. The solution was script, which generalizes the problem so transacting parties can describe their transaction as a predicate that the node network evaluates. The nodes only need to understand the transaction to the extent of evaluating whether the sender's conditions are met.*
>
> *The script is actually a predicate. It's just an equation that evaluates to true or false. Predicate is a long and unfamiliar word so I called it script.*
>
> *The receiver of a payment does a template match on the script. Currently, receivers only accept two templates: direct payment and bitcoin address. Future versions can add templates for more transaction types and nodes running that version or higher will be able to receive them. All versions of nodes in the network can verify and process any new transactions into blocks, even though they may not know how to read them.*
>
> ***The design supports a tremendous variety of possible transaction types that I designed years ago. Escrow transactions, bonded contracts, third party arbitration, multi-party signature, etc.*** *If Bitcoin catches on in a big way, these are things we'll want to explore in the future, but they all had to be designed at the beginning to make sure they would be possible later."*

Satoshi Nakamoto's design fourteen years ago laid the foundation for Bitcoin programming. In the Bitcoin network, there are no "accounts"; there are only "outputs," with the full name being "Transaction Outputs (TXO)". A TXO represents a certain amount of Bitcoin and is the fundamental unit of Bitcoin's system state.

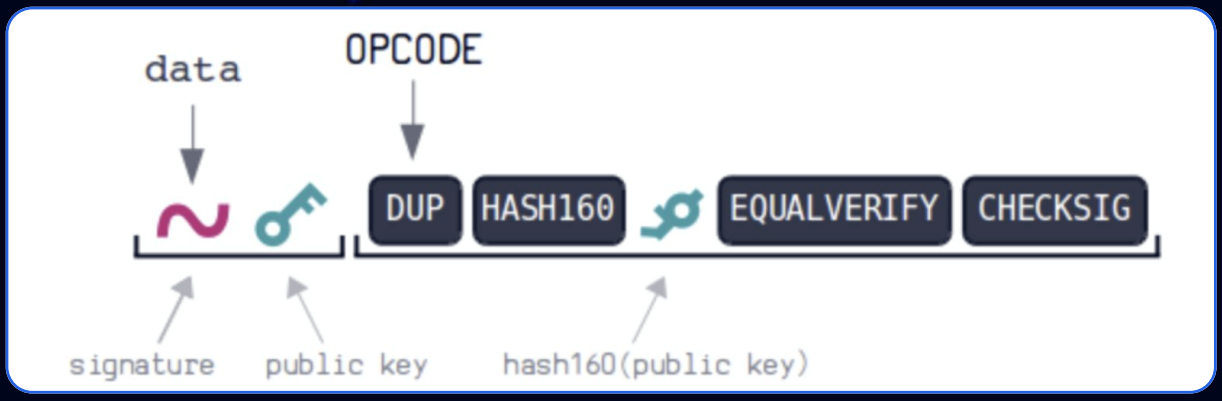
Spending an output means making it an input in a transaction, or providing funds for that transaction. This is why we say the Bitcoin system is based on the "UTXO (Unspent Transaction Output)" model. Only "UTXOs" are the chunks of metal that we can use in the transaction process. When a chunk of metal enters a furnace, it forms new chunks of metal (new UTXOs) after it is burned, and the old chunk of metal, i.e., the "TXO," no longer exists.

Each amount of funds has its own locking script (also called a scriptPubKey) and face value. According to Bitcoin's consensus rules, the scriptPubKey can form a verification program, which is the public key plus specific commands executed in the script—opcodes. To unlock an opcode, a specific set of data called the unlocking script, or scriptSig, must be provided. If the complete script (unlocking script + locking script + opcodes) is valid, the corresponding output will be "unlocked" and can be spent.

Thus, Bitcoin script programming is about programming money and enabling a specific amount of money to respond to particular input data. By designing the scriptPubKey, opcodes, and the interaction process between users, we can offer cryptographic guarantees for the key state transitions of Bitcoin contracts, ensuring the contracts' proper execution.



Here's a simple diagram of a typical P2PKH (Pay to Public Key Hash) script used in Bitcoin:



Source: *https://learnmeabitcoin.com/technical/script/*

If I want to pay Tom 1 BTC, I need to use the spendable UTXO in my wallet to form a UTXO with a face value of 100 million satoshis and include Tom's public key in the UTXO's locking script (along with an operator that checks the signature). This way, only Tom's private key can unlock these funds as the unlocking script corresponding to the signature of his public key.

In summary, Script, or scripting language, is a very basic programming language. It consists of two types of objects: Data—such as public keys and signatures, and opcodes—simple functions that operate on data (a list of opcodes can be found at https://en.bitcoin.it/wiki/Script#Opcodes).

The Arsenal of Bitcoin Programming

Earlier, we mentioned that Satoshi Nakamoto initially envisioned Bitcoin supporting transaction types such as escrow transactions, bonded contracts, third party arbitration, and multi-party signature. So, what weapons are available to implement these, and how are they used in BTCFI?

Multi-party Signature (MULTISIG)

- The form of its locking script is **M <PUB-1><PUB-2>... <PUB-N>N OP_CHECKMULTISIG**, meaning it records n public keys and that m signatures corresponding to these public keys are required to unlock this UTXO.
- For example, two signatures from Alice, Bob, and Chloe (or three public keys) are required to spend the script. The script code is: **2 <Alice> <Bob> <Chloe> 3 OP_CHECKMULTISIG**, where **OP_CHECKMULTISIG** is an opcode that verifies whether the signatures match the provided public keys.

- **Uses include:**

1. **Personal and Corporate Fund Management:** With a 2-of-3 multisig wallet, funds can be accessed as long as two signatures are available. It can also prevent a single wallet manufacturer from doing malicious deeds since multiple manufacturers would need to conspire to withdraw funds.

2. **Transaction Arbitration:**

   - Suppose Alice and Bob want to conduct a transaction, such as purchasing an Ordinals NFT, but cannot make direct payments on delivery. So they agree to lock the funds in a multisig output. Once Bob receives the Ordinals NFT from Alice, he can release the entire funds to her.

     To prevent cases where payments are not made upon the receipt of goods, a third party could be introduced, forming a 2-of-3 multisig output. In case of a transaction dispute, the third party can be the judge. If the third party believes Alice has delivered the goods, they can work with her to transfer the funds.

   - As long as the third party (e.g., he or she is an oracle) publicly shares their public key, the two parties to the transaction can include it in the 2-of-3 multisig script, thus incorporating the arbitrator. It can be done without the arbitrator's knowledge because the output on the blockchain records the hash value of the script.

     However, the problem here is that the third-party oracle can determine the specific outcome of the contract, which introduces certain risks. Discrete log contracts (DLCs) mentioned later in this article have made optimizations regarding this issue so that this approach can truly be used for BTCFI applications like Bitcoin lending.

Timelocks

Timelocks control the validity of a transaction and when an output can be spent. They are Bitcoin scripting weapons essential for BTCFI scenarios such as staking, restaking, and collateralized lending. Developers must choose between relative timelocks (nSequence) and absolute timelocks (nLocktime):

- Absolute Timelock (nLocktime): A transaction is only considered valid and can be included in a block after a specific time. The script-level absolute timelock uses the **OP_CLTV** opcode, which verifies that the UTXO can only be unlocked after a specific time. For example, this money can only be spent after a block height of 400,000.
- Relative Timelock (nSequence): This means the transaction becomes valid only after a certain period after the inputs of the transaction that created the UTXO (i.e., the prior transaction) are confirmed on the blockchain. The script-level relative time lock uses the **OP_CSV** opcode. For example, this money can only be spent 3 blocks after block confirmation.

Hashlocks (Hash Preimage Verification)

In addition, there are also hashed timelocks that combine hashlocks (hash preimage verification), which are often used in **Bitcoin staking and restaking:**

- The form of a hashlock's locking script is OP_HASH160 <hash> OP_EQUAL, which verifies whether the data in the unlocking script is the preimage of the hash in the locking script.
- Hashed Timelock Contract (HTLC): If the recipient of the funds cannot provide the preimage of a hash within a specified time, the funds can be recovered by the payer.

Flow Control (Parallel Unlocking Conditions)

The **OP_IF** opcode can be used in locking scripts to set up multiple unlocking paths, allowing the UTXO to be unlocked as long as any path's conditions are met. The hashed timelock contract mentioned earlier also utilizes such flow control opcodes.

After the Taproot upgrade, the MAST (Merkelized Abstract Syntax Tree) feature allows us to place different unlocking paths in different leaves of a Merkle tree. Babylon's BTC staking transactions also use MAST, which enhances transaction efficiency and privacy, which we'll cover later.

SIGHASH: Signature with Additional Information

SIGHASH flags can be used when signing Bitcoin transactions. These flags specify additional conditions for the validity of the signature, allowing us to modify parts of the transaction without invalidating the signature. They reflect the signer's intent or delegation for the use of the signature. For example:

- **SIGHASH_SINGLE | ANYONECANPAY** signs the input and the output that shares the same index number as the input, while other inputs and outputs can be altered without invalidating the signature. Andy can sign an input worth 1 BTC and an output worth 100 Runes so that anyone willing to exchange 100 Runes for 1 BTC can complete the transaction and have it processed on the blockchain.
- Relative Timelock (nSequence): This means the transaction becomes valid only after a certain period after the inputs of the transaction that created the UTXO (i.e., the prior transaction) are confirmed on the blockchain. The script-level relative time lock uses the **OP_CSV** opcode. For example, this money can only be spent 3 blocks after block confirmation.

Another example is the Schnorr signature introduced after the Taproot upgrade, which can be used for Discreet Log Contracts (DLCs).

Limitations of Bitcoin Programmability

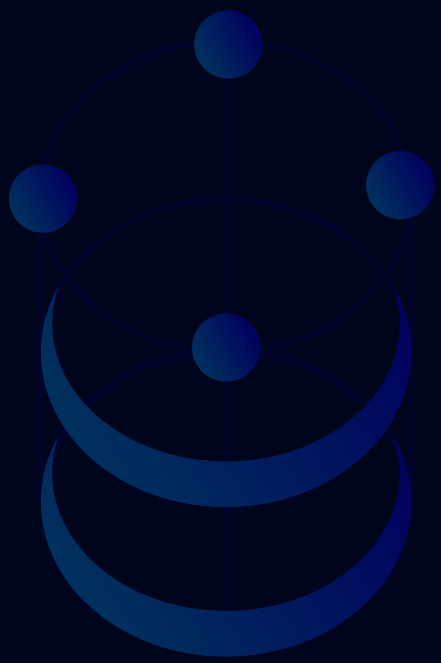
The basic model of Bitcoin programming is where UTXO locking scripts specify verification conditions, unlocking scripts provide data, and opcodes in locking scripts indicate the verification program (such as signature verification, hash preimage verification, etc.). The funds can be spent once the verification program is passed. The core limitations are as follows:

1. Only a few verification programs are available: Implementing zero-knowledge proofs or other verification programs entails a fork. Therefore, although the BTCFI expansion plan Fractal proposed by Unisat maintains 100% consistency with Bitcoin, it deviates from the Bitcoin mainnet in terms of liquidity and security in order to implement proposals of "controversial" opcodes like OP_CAT and ZK-native verification opcodes.

2. **Bitcoin scripts have no computational power:** As long as funds can be unlocked, any path can grant access to the full amount. How the funds are spent cannot be restricted after unlocking, which means it is **difficult to adopt variable interest rate schemes in BTCFI lending projects and only fixed rates are feasible.** To address this issue, the Bitcoin community is discussing implementing "covenants", which could unlock more BTCFI application scenarios by restricting further spending on transactions. BIP-420, OP_CAT, OP_CTV, APO, OP_VAULT, and others mentioned by Taproot Wizard are related to this.
3. **UTXO unlocking conditions are completely independent:** A UTXO cannot decide whether it can be unlocked based on the existence or conditions of another UTXO. This issue frequently arises in BTCFI collateralized lending and staking. Partially Signed Bitcoin Transactions (PSBTs) discussed later are used to solve this problem.

Adjustments and Evolution of Bitcoin Contracts

While Ethereum contracts are computation-based, Bitcoin contracts are verification-based. This stateless contract model has brought us many challenges in developing BTCFI products. Over the ten years or so of developing Bitcoin contracts, the ingenious use of cryptographic algorithms and signatures has significantly enhanced privacy, efficiency, and decentralization, making BTCFI products possible.



Discreet Log Contracts (DLCs): Solving Decentralized Liquidation Issues in BTCFI Scenarios

When it is necessary to liquidate users' positions based on the oracle prices in lending, options, and futures contracts, it is unavoidable to retain operational control over user assets, which undoubtedly introduces trust costs for users to trust the protocols to act in good faith.

The Discreet Log Contract (DLC) is introduced to address this issue. It uses a cryptographic technique known as adaptor signatures to allow Bitcoin scripts to program **financial contracts dependent on external events while fully ensuring privacy.**

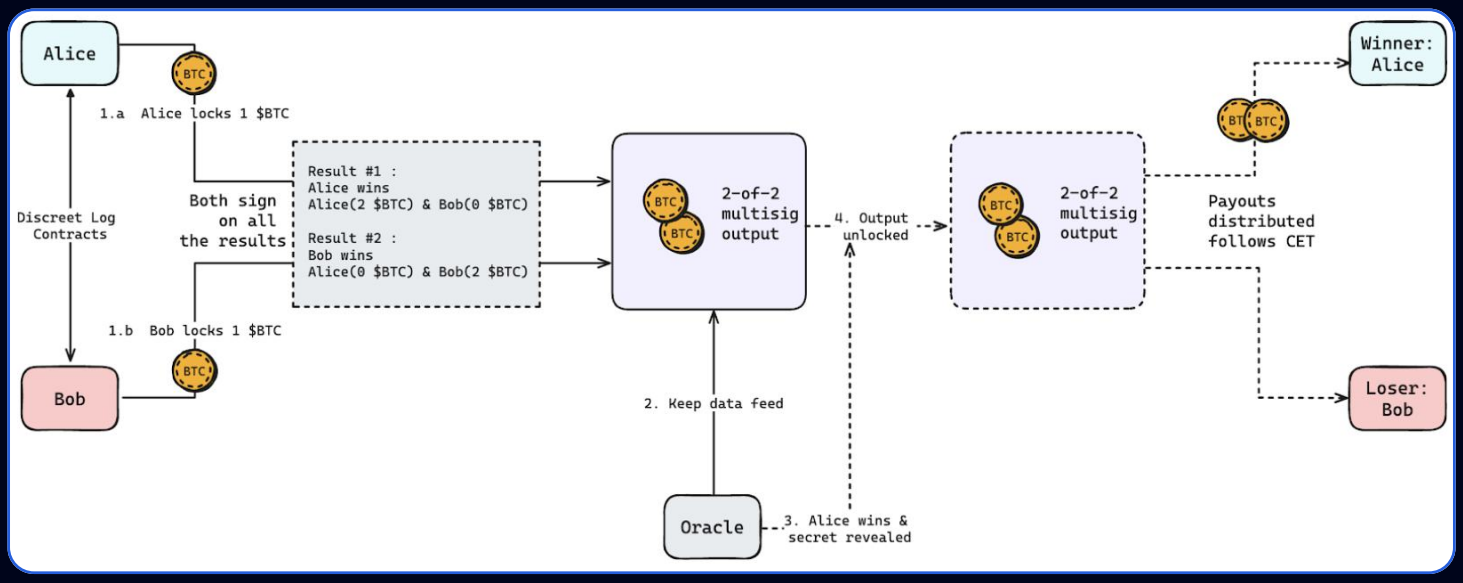
DLCs were proposed in 2017 by Tadge Dryja (a research scientist) and Gert-Jaap Glasbergen (a software developer) at MIT's Digital Currency Initiative and were publicly demonstrated on March 19, 2018.

Adaptor signatures enable a signature to become valid only after a private key is added. The Schnorr signature introduced in the Taproot upgrade is an example of an adaptor signature.

Generally speaking, the standard form of a Schnorr signature is **(R, s)**. Given **(R, s')**, as long as x, i.e., the secret, is known, we can set **s = s' + x**, which gives **(R, s)**. Here's a simple example to explain its application:

- Alice and Bob each bet 1 $BTC on opposite outcomes of a sports match (assuming no draw). The winner will take the entire 2 $BTC stake. They lock the stake in a multisig wallet that requires multi-party signatures to release the funds.
- Choose the oracle. It will publish the match result (usually sourced off-chain, such as from an exchange or sports website).
- The oracle does not need to know any details about the bet, not even who is participating in the DLC. Its role is strictly limited to providing the result. Once the event occurs, the oracle issues a cryptographic proof called a commitment, indicating that it has determined the outcome.

- To claim funds locked in the multisig wallet, the winning party, Alice, uses the secret provided by the oracle to create a valid private key, enabling them to sign a transaction to spend the funds in the wallet.
- This transaction is then added to the Bitcoin blockchain for settlement. Since the signature is an ordinary one, the contract does not appear to be a Discreet Log Contract (DLC). This is totally different from other common multisig models, where the contract's content is fully disclosed and the oracle directly participates in decision-making.



Source: *https://shellfinance.gitbook.io/shell*

Liquidation Mechanism in Lending Protocols

Assume Alice uses $ORDI as collateral and borrows an amount equivalent to 0.15 $BTC. The loan position will become pending liquidation when and only when the oracle quotes $ORDI below 225,000 sats/ORDI. DLC allows liquidators to liquidate the position without permission in the pending liquidation state while ensuring that users' collateral cannot be manipulated when the price does not reach the liquidation price. In this scenario, Alice effectively places a bet with the lending protocol on the price of $ORDI:

- If the price falls below 225,000 sats/ORDI, the protocol can obtain all of Alice's collateral and assume the corresponding BTC debt;
- If the price is higher than or equal to 225,000 sats/ORDI, nothing happens, and ownership of the assets remains unchanged.

Then, we need the oracle to commit to publishing the signature of the result **s_c_1** with a nonce **R_c** when the price is below 225,000 sats/ORDI:

- Alice and the protocol only need to establish a commitment transaction for the result and create an adaptor signature **(R, s')** rather than a signature **(R, s)**, which means that the signatures exchanged between the parties cannot directly unlock the contract and a secret must be revealed. This secret is the preimage of **s_c_1.G**, that is, the oracle's signature. Since the oracle's signature nonce is already determined, **s_c_1.G** can be constructed.
- When the price drops below 225,000 sats/ORDI, the oracle will release the signature **(R_c, s_c_1)**. The protocol can then complete the counterparty's adaptor signature and add its own signature, making the above transaction valid and broadcasting it to the network to trigger the settlement.
- Conversely, if the price does not fall below 225,000 sats/ORDI, the oracle will not release **s_c_1,** and the commitment transaction cannot become valid.

In essence, DLC allows users and protocols to make agreements as participants using the Bitcoin blockchain, where both parties lock their funds in a multisig address to construct a DLC script. These funds can only be used and redistributed following specific rules if the oracle publishes specified information at a predetermined time.

This way, the lending protocol, with the help of DLC, achieves a liquidation mechanism involving an external price oracle, without requiring users to trust any entity.

We will later discuss how lending protocols like Liquidium and Shell Finance utilize this technology for permissionless, decentralized liquidation.

Oracles' Roles

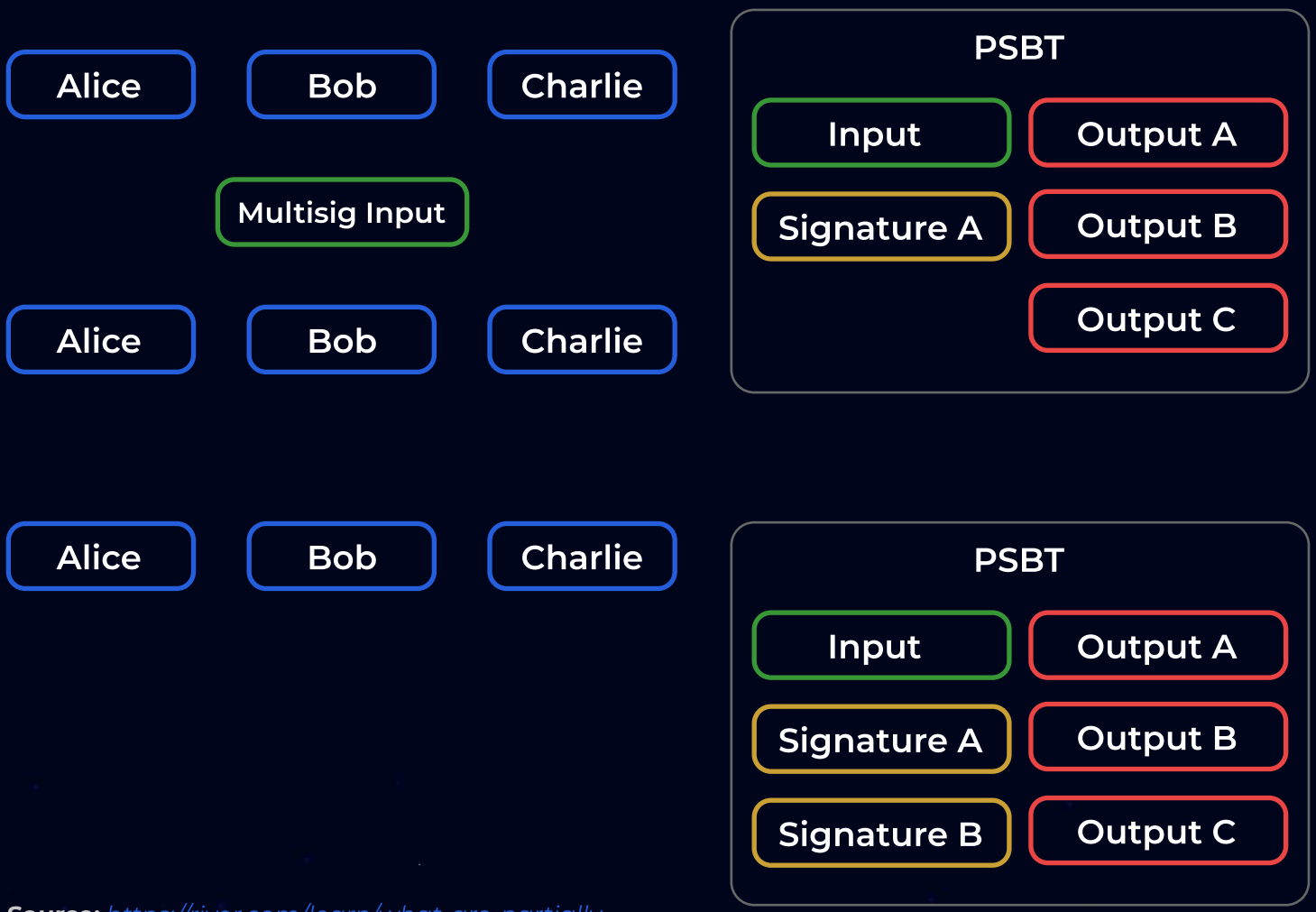
In DLCs, oracles provide price feeds at a fixed frequency and participate in the process of generating and revealing secrets within the DLC mechanism as third parties.

At present, DLC oracles are not yet standardized products. Lending protocols are the primary ones involved in developing DLC modules, while standardized oracles like Chainlink provide off-chain data feeds. However, as DLC-based lending protocols are launched and continue to develop, many existing oracle projects are exploring how to develop DLC oracles.

Partially Signed Bitcoin Transactions (PSBT): Making Custody of Funds Unnecessary in Multi-Party Transactions in BTCFI Protocols

PSBT comes from Bitcoin standard BIP-174, which allows multiple parties to sign the same transaction in parallel and then combine their respective PSBTs to form a fully signed transaction. The multiple parties here can be protocols and users, buyers and sellers, stakers and staking protocols, etc.

Alice, Bob, and Charlie have funds stored in a 2-of-3 multisig. They wish to withdraw and split the funds into three. The three of them must sign the same transaction to spend the UTXO. If they do not trust each other, how can they ensure the security of the funds?



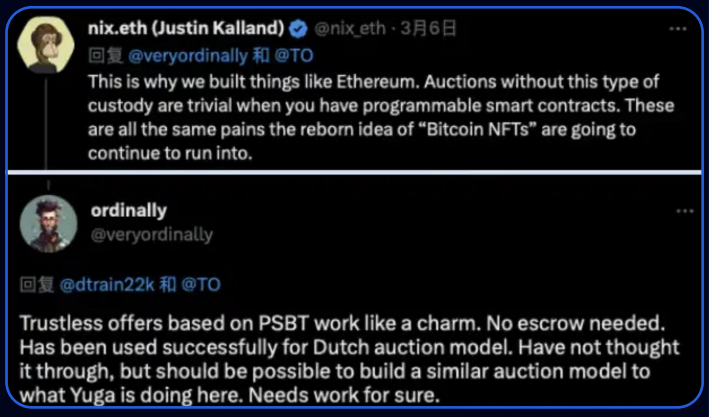
Source: *https://river.com/learn/what-are-partially-signed-bitcoin-transactions-psbts/*

- First, Alice creates a PSBT as the initiator with the multisig UTXO as the input and adds the wallet addresses of the three as outputs. Since PSBT ensures that no transaction other than this particular one can use any of the signatures, Alice can sign it and then send it to Bob.
- Similarly, if Bob finds no issues after he checks the PSBT, he signs it as well; after signing, he passes it to Charlie for signing and broadcasting the transaction. Charlie does the same.

Therefore, "partially signed" means that each person only needs to check the part of the transaction that concerns them. As long as this part is correct, it ensures that the transaction will proceed without issues once it is processed on the blockchain.

On March 7, 2023, the auction for Yuga Labs' Ordinals NFTs adopted a highly centralized custodial auction model. During the auction, all funds for the auction were required to be deposited into Yuga's custodial address, which put funds at a severe security risk.

Users of the Ethereum ecosystem pointed out that Yuga's auction highlighted the importance of ETH smart contracts, but Ordinals developers responded that trustless offers based on PSBT work like a charm and can facilitate NFT transactions between buyers and Yuga Labs without the need for escrow.



Source: *https://x.com/veryordinally*

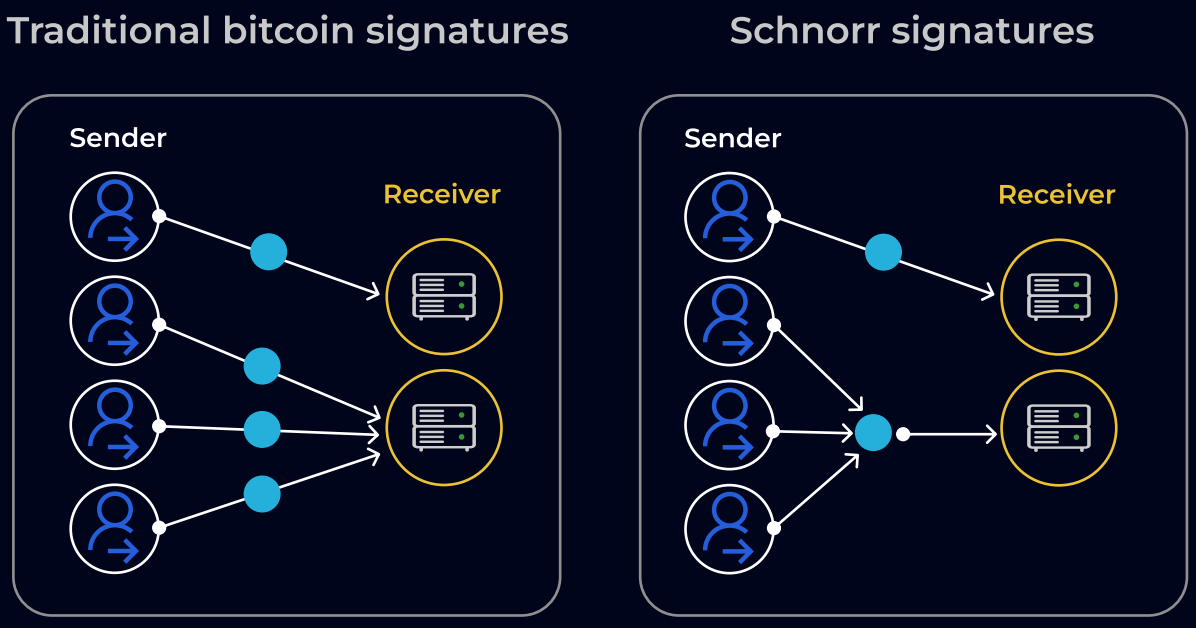
Assume now there is a buyer and a seller to a Bitcoin NFT transaction, and the NFT seller's public key is known to both parties. When initiating an NFT transaction, the buyer first writes their UTXO input and an output to receive the NFT. After setting up the transaction and signing it, the buyer converts it to PSBT and sends it to the seller. The seller, upon receiving the message via the protocol, signs it, and the Bitcoin NFT transaction is completed.

The entire process above is completely trusted by both buyer and seller. For the buyer, the bid, receiving address, and other information are pre-set in the transaction, and any changes will invalidate the signature. For the seller, the NFT is only sold when they complete the signature, and the price is determined after their own evaluation.

Taproot Upgrade: Unlocking the Pandora's Box for Bitcoin Ecosystem and BTCFI Boom

The Taproot upgrade was activated in November 2021, aiming to enhance Bitcoin's privacy, improve transaction efficiency, and expand Bitcoin's programmability. Through the implementation of Taproot, Bitcoin can now hold in custody large-scale smart contracts with tens of thousands of signatories while concealing all participants and maintaining the size of a single-signature transaction. This makes more complex on-chain BTCFI operations possible. Almost all BTCFI projects have adopted the scripting language of the Taproot upgrade.

1. **BIP340 (BIP-Schnorr):** It supports multi-party signatures for a single transaction. When used in the aforementioned Discrete Log Contract (DLC) applications, a transaction can only be executed under predefined conditions. The amount of data committed to Bitcoin is the same as that of a standard single-signature transaction.

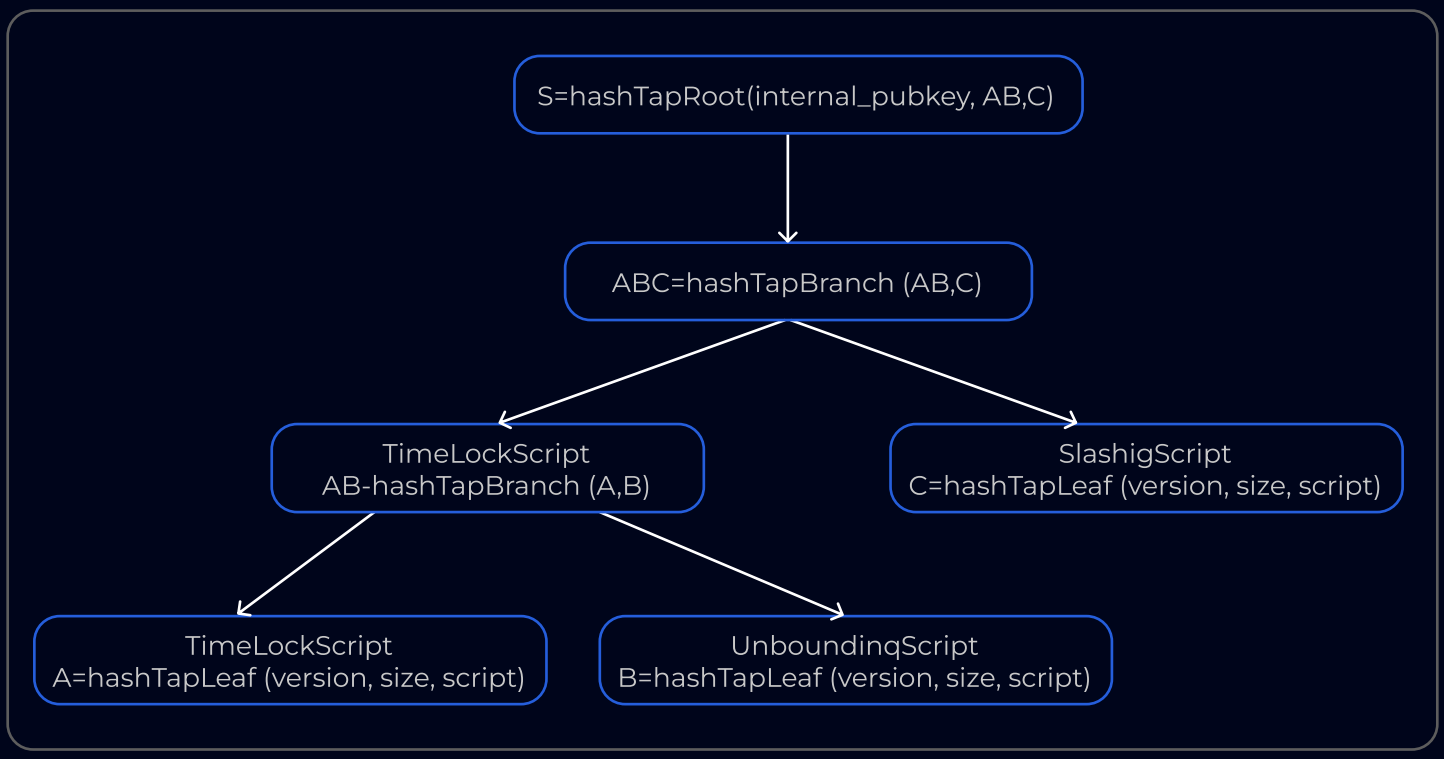


Source: *https://cointelegraph.com/learn/a-beginners-guide-to-the-bitcoin-taproot-upgrade*

2. **BIP341 (BIP-Taproot):** Taproot introduces Merkle Abstract Syntax Trees (MAST), committing fewer contract transaction data on-chain. This enables Bitcoin to create more complex contracts. Unlike existing Pay-to-Script-Hash (P2SH) transactions, MAST allows users to selectively reveal parts of the script as needed, thus improving privacy and efficiency. MAST has worked excellently in Babylon's BTC staking transactions, constructing multiple locking scripts into a single transaction with multiple scripts. The three locking scripts are as follows:

- **TimeLockScript:** It enables the lock-in function for staking;
- **UnboundingPathScript:** It allows for the early termination of staking;
- **SlashingPathScript:** It implements a slashing function for malicious behavior.

These scripts are all leaf nodes, from which a binary tree is gradually constructed as follows:

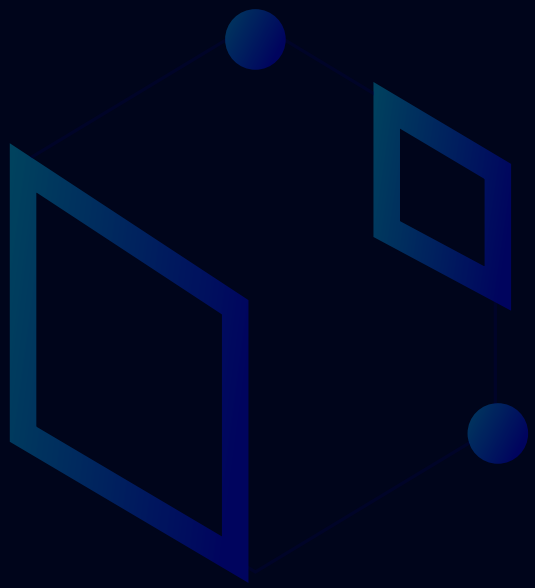


Source: *https://blog.csdn.net/dokyer/article/details/137135135*

3. **BIP342 (BIP-Tapscript):** It delivers an upgraded transaction programming language for Bitcoin, which leverages Schnorr and Taproot technologies. Tapscript also allows developers to realize future Bitcoin upgrades more efficiently.

4. Laying the Foundation for the Ordinals Protocol:

- The Taproot upgrade also introduces the Taproot (P2TR) address, which starts with bc1p, so that metadata is written into the spent script stored in the Taproot script path but never appears in the UTXO set.
- Since maintaining/modifying the UTXO set requires more resources, this approach can save huge amounts of resources by increasing the data storage capacity of one block, which means there is now space to store images, videos, and even games, thus unintentionally paving the way for the deployment of Ordinals. The inscription addresses we commonly use are Taproot (P2TR) addresses.
- Since Taproot script spends can only be made from existing Taproot outputs, inscriptions are made using a **two-phase** commit/reveal procedure for minting. First, in the commit transaction, a Taproot output committing to a script containing the inscription content is created. Second, in the reveal transaction, the transaction is initiated with the UTXO corresponding to the inscription as input. At this point, the content of the corresponding inscription is made public to the entire network.
- As new assets like Ordinals BRC-20, ARC-20, and Runes emerge, Taproot's adoption rate for transfers is kept at around 70%.



Ordinals and BRC-20: Creating Blue-Chip Assets for BTCFI and Opening the Door Based on Indexer Programming

Ordinals have realized the dream of Bitcoin OGs to shop on the Bitcoin mainnet, with their market value surpassing that of Ethereum NFTs.

- Ordinals were proposed by Bitcoin core contributor, Casey Rodarmor, in January 2023. The core concept is the theory of ordinals, aiming to assign each satoshi (sat), the smallest unit of Bitcoin, a unique identifier and attribute, so as to transform it into a unique non-fungible token (NFT). By inscribing various types of data (images, text, videos, etc.) into satoshis, the Ordinals protocol realizes the creation and trading of Bitcoin NFTs.
- This process not only expands the uses of Bitcoin but also allows users to directly create and trade digital assets on the Bitcoin blockchain. The lasting value lies in the fact that Ordinals are created based on Bitcoin's satoshis, so their fundamental value is linked to Bitcoin itself, which means theoretically their values will not be reduced to zero.

BRC-20 is a token system for on-chain recording and off-chain processing that uses JSON data's ordinal inscriptions to deploy token contracts, mint tokens, and transfer them.

- It treats inscriptions as an on-chain ledger for recording the deployment, minting, and transfer of BRC-20 tokens.
- In settlements, it is essential to resort to off-chain queries and rely on third-party indexing tools to search Bitcoin blocks and record all deployment, minting, and transfers of BRC-20 tokens to determine the final balance of each user's BRC-20 tokens. This may lead to different query results for the balance of the same account on different platforms.

Ordinals and BRC-20 not only provide trading demand and blue-chip assets for BTCFI but also offer many BTCFI projects new ideas based on indexer programming that enhance Bitcoin's contract capabilities. The "op" field combination of JSON can further evolve into inscription-based DeFi, SocialFi, and GameFi. The approach based on indexer programming is used by **AVM, Tap Protocol, BRC-100, Unisat's swap function, and many other projects that propose to build smart contract platforms on Bitcoin's layer 1**

MuSig2: Decentralized Model for Exploring Bitcoin Restaking and LST

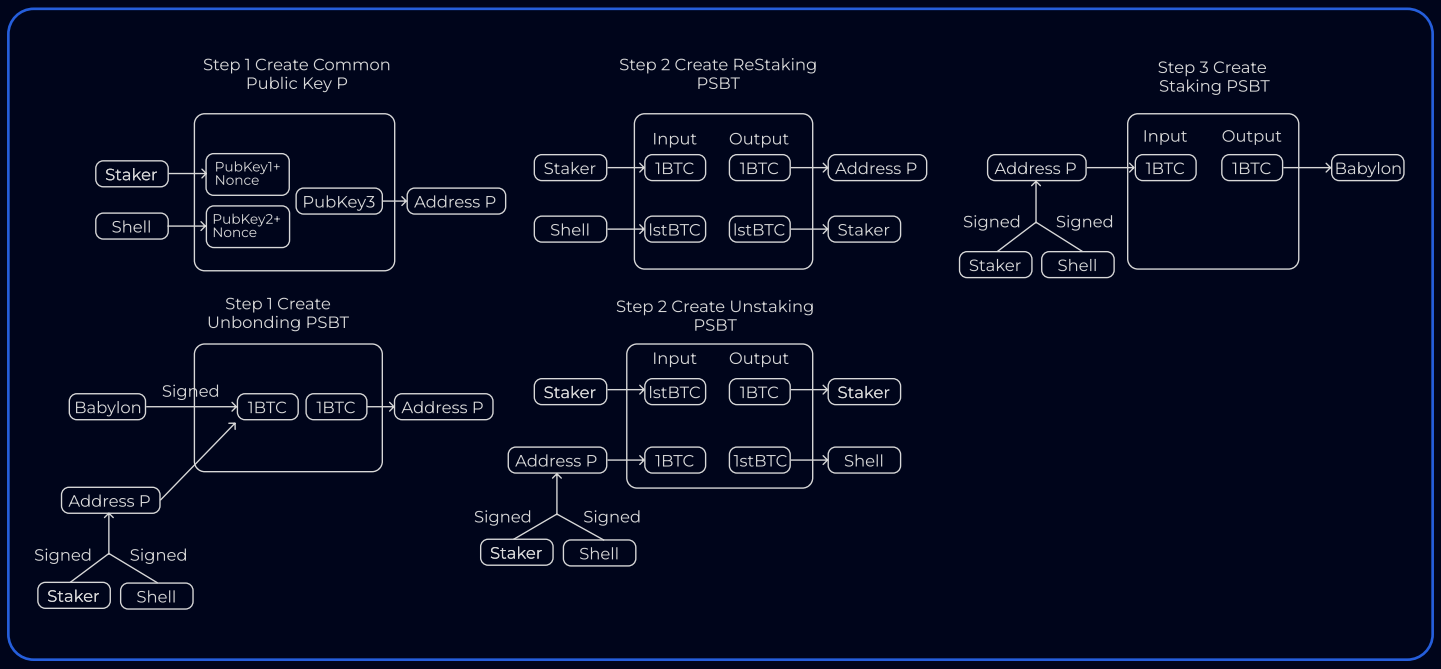
Multi-signature schemes allow a group of signers to jointly sign on a message. MuSig allows multiple signers to create an aggregated public key from their **respective private keys and then collectively produce a valid signature for that public key. It is an application of the Schnorr signature. As mentioned earlier, the standard form of a Schnorr signature is (R, s). Given (R, s'), as long as x, i.e., the secret, is known, we can set s = s' + x, which gives (R, s). The aggregated public key and the valid signature here are generated using the private key and a nonce.**

Only two rounds are needed in the MuSig2 scheme to complete a multi-signature. The aggregated public key created this way is indistinguishable from other public keys, which enhances privacy and significantly reduces trading fees. The Taproot upgrade is compatible with the MuSig2 multi-signature scheme, and its BIP proposal was published in the 2022 Bitcoin BIP-327: MuSig2 for BIP340-compatible Multi-Signatures.

While liquid staking on Ethereum can be achieved through smart contracts, Bitcoin lacks the contract capabilities needed for liquid staking. As previously mentioned, Bitcoin whales generally detest centralized custodians and require MuSig2 for decentralized Bitcoin liquid staking. Here, we use the Shell Finance solution as an example.

1. The user and Shell Finance compute an aggregated public key and the corresponding MuSig2 multisignature “address P” based on their respective private key data and two nonces generated by the wallets.
2. In the PSBT transaction constructed by Shell Finance, the assets of users and Shell Finance are staked from MuSig2-supported multi-signature “address P” into Babylon, and the wallet again provides nonce support, feeding it into the aggregated public key corresponding to the multi-signature address.
3. When the Babylon staking period ends, Shell Finance constructs a PSBT unlock transaction, and both the user and Shell Finance sign to unlock the staked assets.

A third-party wallet that generates the nonce, the staking user, and the LST project party together create the aggregated public key and signature. In this process, both the user and the project party only have custody of a private key, and without the nonce value, they cannot generate the aggregated public key or signature to retrieve the funds; meanwhile, the wallet cannot move the funds without the private key. If the nonce is generated by the project party alone, there is a risk that the project party will take malicious behavior, so users should be cautious.



Current Application Scenarios of BTCFI

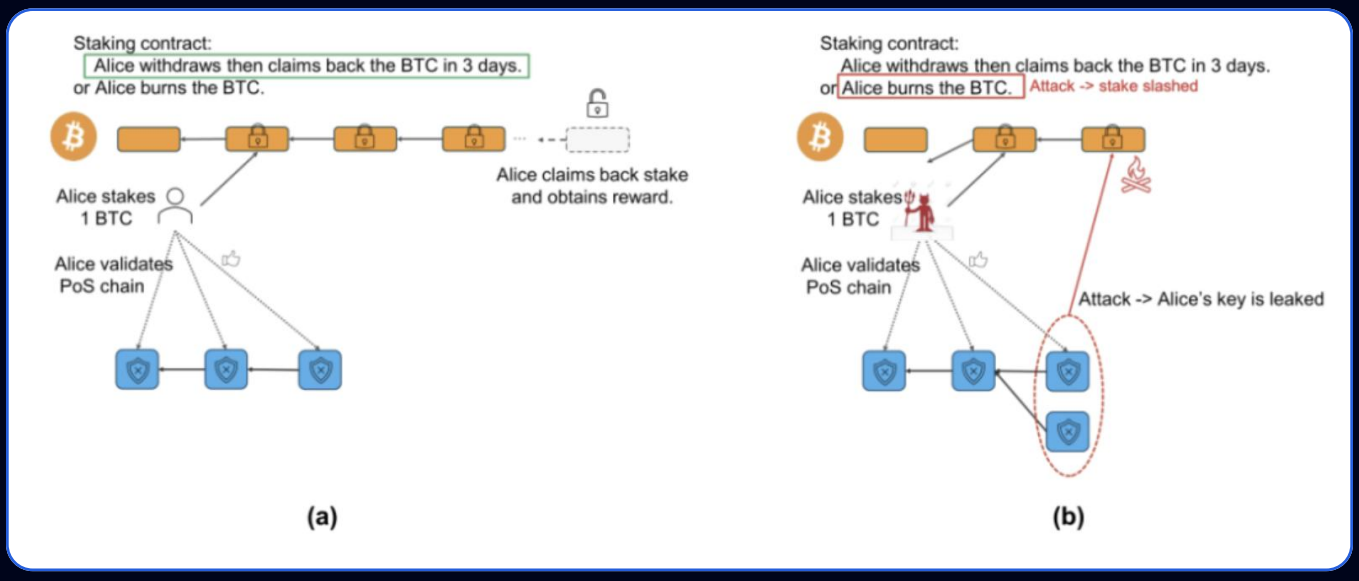
Bitcoin programming is not complicated and is in fact much simpler than languages like Rust. The key lies in creating verifiable, credible commitments while offering technical security superior to that of Ethereum. This defines the boundaries for BTCFI development. The greatest challenge is which BTCFI products with PMF (project market fit) can be developed within these boundaries.

Just as developers didn't realize that they could use Solidity contracts on Ethereum to develop the x*y=k AMM algorithm when these contracts were first launched, they started by exploring directions such as ICOs, order books, and peer-to-peer lending.

Liquidity Boost: Babylon — The Catfish in BTCFI

Babylon has built a set of trustless staking protocols with no intermediaries that allow direct staking of Layer 1 Bitcoin to earn yields. It also extracts Bitcoin's security and shares it with POS chains, providing POS security guarantees for Cosmos and other Bitcoin Layer 2s as a universal shared security layer, thus sharing in the Bitcoin economic security.

- **Absolute Security:** BTC staking has one significant advantage over other staking forms, which is that when the protected POS chain is attacked and collapses, the staked Bitcoin remains unaffected. Specifically, if a POS chain is attacked and its token value drops to zero, users holding tokens on the POS chain would suffer losses. However, with BTC staking, even if the protected POS chain is attacked and fails, the user's Bitcoin remains secure.
- **Slashing Mechanism:** If a user engages in malicious behavior such as double-signing on a POS chain from which Babylon borrows security, the corresponding assets can be unlocked using EOTS (extractable one-time signatures), and a portion of them will be forcibly sent to a burn address by the network's executing roles.



Source: *https://docs.babylonchain.io/papers/btc_staking_litepaper(EN).pdf*

The Babylon mainnet is now live, with 1,000 BTC staked in the first phase. The second phase will be launched soon.



Finality Provider	BTC PK	Total Delegation	Commission
RockX	bb0b...fdd1	297.82656 BTC	3%
Solv Protocol	fa74...8bfb	150.40155 BTC	5%
Lorenzo	db91...9923	129.3655 BTC	5%
PumpBTC	0f5c...85e4	118.4288 BTC	5%
OKX Earn	3746...032c	114.94999767 BTC	3%

Source: *https://btcstaking.babylonlabs.io/*

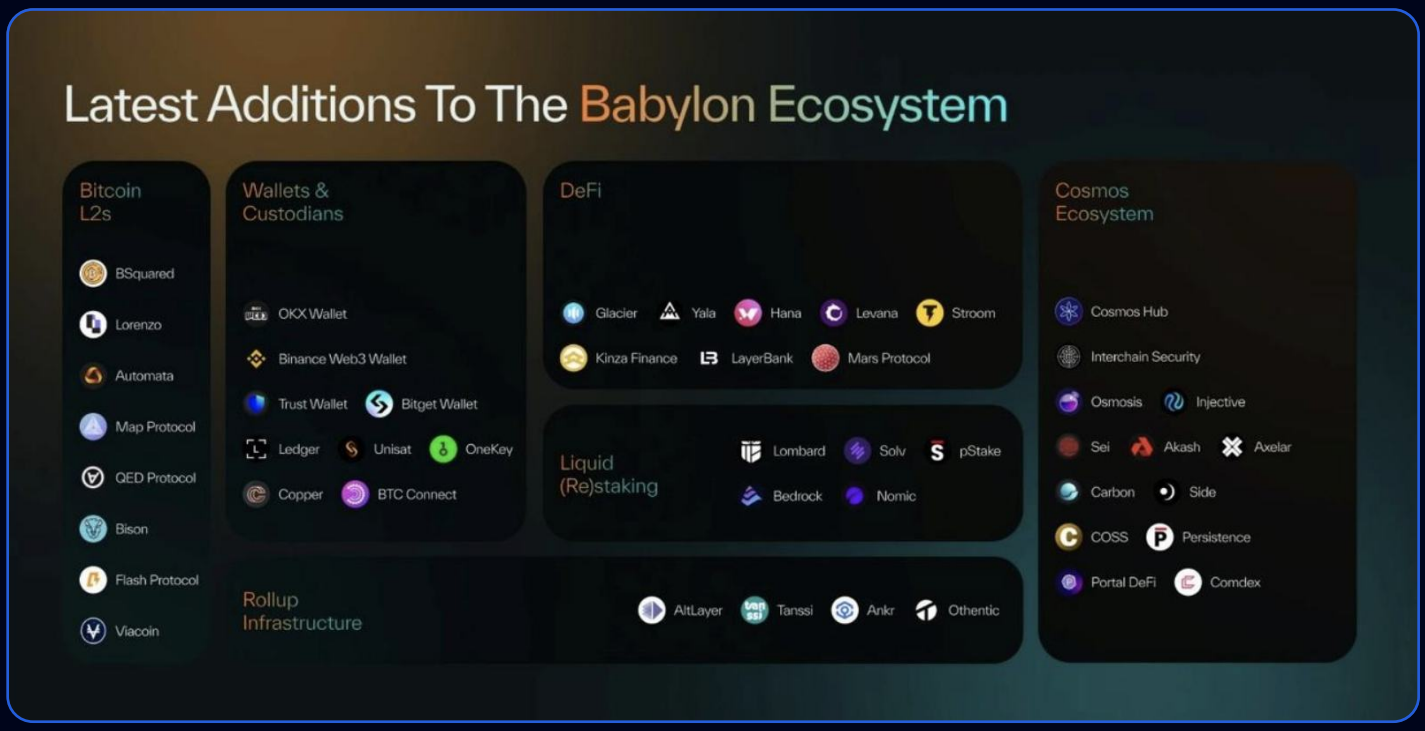
In the first phase, large holders contributed the majority of staked BTC, with their gas fees paid making up 5% of the total. The second and third phases may attract participation from more retail investors.

Attracting Massive BTC into BTCFI Staking for the First Time:

- Although Babylon cannot offer ETH-margined returns from POS like Ethereum, its 3-5% APY is attractive enough to large, lazy Bitcoin holders and miners without expectation of high returns who prioritize security and are reluctant to use cross-chain wrap solutions, as well as to European, American, and Asian funds that are optimistic about the Bitcoin ecosystem. Therefore, when the total deposit reaches 100,000 BTC, an equivalent of $100 million in token returns or more would suffice.
- Babylon is currently collaborating actively with the Cosmos ecosystem, including well-known projects like **Cosmos Hub, Osmosis, and Injective.** It allows these projects to become AVS in the future and offer their tokens as rewards to Bitcoin restakers, which further raises Babylon's BTC deposit cap.

Babylon Injects Significant Liquidity into the BTCFI Ecosystem, Educates Users, and Invigorates the Ecosystem.

- The Ethereum ecosystem once saw quite successful narratives related to restaking like DeFi and Layer 2. On the other hand, it is the first time that Babylon has allowed staking for yields on the Bitcoin mainnet. Most Bitcoin holders are reluctant to take risks to engage in custody and cross-chain solutions, so this is equivalent to their first experience with BTCFI. In addition, they may further explore other options like LSTs.



- In the LST sector of the Babylon ecosystem alone, dozens of projects have emerged, including StakeStone, Uniport, Chakra, Lorenzo, Bedrock, pSTAKE Finance, pumpBTC, Lombard, and SolvBTC, along with various other DeFi projects. Bitcoin ecosystem projects that struggle to secure initial TVL can leverage Babylon's BTC staking to attract BTC through LSTs, and the LST assets can also be utilized within their ecosystem.
- Since Babylon generates returns in the form of tokens rather than BTC/ETH, it is less appealing to major players. Therefore, overall, it will not be as centralized as ETH staking. Instead, since its tokens generate uncertain amounts of yields, early-stage start-ups that can explore unconventional paths have the opportunity to carve out their own niches.

Multiple Blue-Chip LST Assets May Emerge on the Bitcoin Mainnet and Give Rise to BTCFI Demand

Babylon has pioneered a new sector of native BTC staking for yields, offering a large-scale application scenario for the previously dormant hundred trillions of BTC on the mainnet. The mass staked BTC has given rise to numerous liquidity staking tokens (LSTs).

These staking certificates, derived from BTC, become natural blue-chip collateral for collateralized lending and other scenarios, thereby creating the conditions for the development of lending based on Bitcoin's native assets, stablecoins, and swaps, i.e., BTCFI.

- The core reason why BTCFI development has been difficult is that there has long been an absence of high-quality assets on the Bitcoin mainnet, which has directly led to a shortage of collateral for lending, a lack of trading demand for swaps, and shallow liquidity pools. Currently, the only blue-chip assets on the Bitcoin mainnet are SATS and ORDI in BRC-20 and node monkey of Ordinals NFTs, among others.
- However, there will be favorable conditions for the development of BTCFI if a portion of the staked assets in Babylon can generate liquidity staking tokens, similar to how Lido issues stETH on Ethereum, be used as collateral for lending protocols like Aave and Compound, and form significant trading depth on Uniswap.
- There may be many stakers who wish to borrow BTC using liquidity staking tokens, either to reinvest in staking or to hedge against risks.



Innovation in Asset Issuance: Two Major DEXs, Unisat and Magic Eden, Are Set to Launch



Source: *https://docs.unisat.io/knowledge-base/brc20-swap-introduction*



Source: *https://magiceden.io/runes/DOG%E2%80%A2GO%E2%80%A2TO%E2%80%A2THE%E2%80%A2MOON*

- Unisat's BRC-20 swap will launch in September. It also supports Runes by mapping them to BRC-20 tokens. Subsequently, tokens can be issued and traded by adding liquidity pools, no longer needing to raise gas prices for minting tokens or trade token inscriptions like NFTs one by one, thus enabling batch transactions.
- Magic Eden's DEX for Runes is also scheduled to go online in Q4 this year.

A Fully BTC-Native Peer-to-Pool Lending Protocol for Stablecoins Is About to Launch

Liquidium is a lending protocol entirely built on the Bitcoin mainnet, realized through partially signed Bitcoin transactions (PSBTs) and discrete log contracts (DLCs) as mentioned earlier. Specifically:

- Lenders fill out an offer, including metrics such as LTV (the ratio of the loan amount to the value of the collateral), interest, and floor price, and then deposit Bitcoin.
- Borrowers select a lender from the platform's offers and deposit NFT or Rune assets.

Launched in October 2023, the protocol has registered a trading volume of 2,227 BTC in less than a year, indicating that there is demand for BTCFI lending on the Bitcoin mainnet.

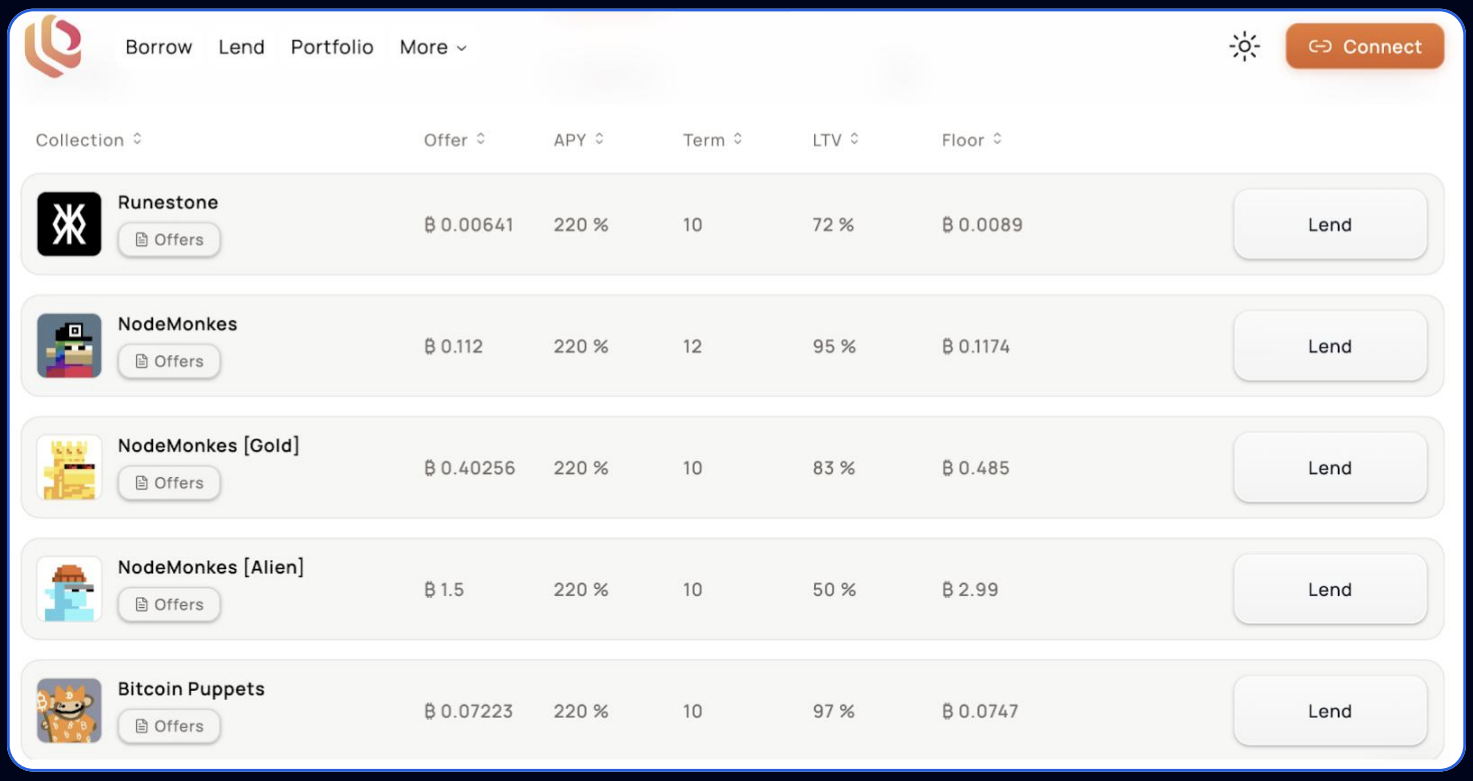


Source: *https://dune.com/shudufhadzo/liquidium*

Core issues lie in:

1. **Low Fund Utilization Efficiency:** If no borrowers voluntarily take the offer, the lender's Bitcoin remains idle. Canceling and placing orders also incur fees. In other words, the protocol is incapable of order matching and has a discovery process.
2. **Peer-to-Peer Liquidation:** Liquidators here are only the lender and borrower, with no others involved.

- Once the value of the NFT or Runes falls below the lending amount relative to the LTV, the borrower is likely to default. In this case, the one offering the offer is left with the NFT or Runes, effectively bearing the risk of the downturn.
- From another perspective, if the borrower's NFT or Runes declines in value, he or she must either repay the loan immediately or lose the NTF or Runes, which is also unfair to the borrower.
- To prevent borrowers from defaulting, the term is limited to just over ten days, with a very high APY.



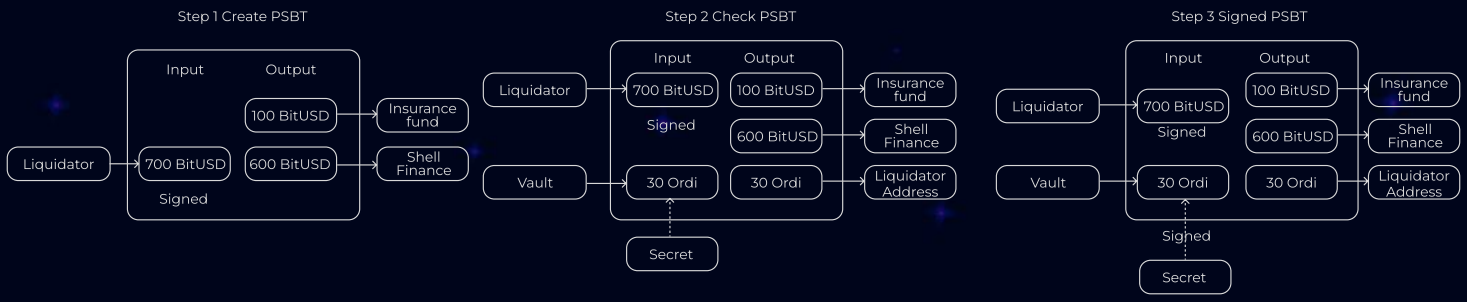
Source: *https://liquidium.fi/*

This may be why AAVE's predecessor, Ethlend, struggled to develop sustainably, as peer-to-peer lending is too difficult to scale consistently.

Shell Finance maximizes liquidity aggregation in lending and liquidation scenarios using its synthetic stablecoin, $bitUSD, so as to realize a Bitcoin version of peer-to-pool lending. The positive cycle of borrowing and repaying with $bitUSD is expected to produce strong scale effects in the future.

In the liquidation and transaction construction processes, discreet log contracts (DLCs) and partially signed Bitcoin transactions (PSBTs) are also used to achieve non-custodial and decentralized liquidation of collateral and funds. Specifically:

- Borrowers can stake Ordinals NFTs, BRC-20 tokens, Runes, and other assets (other assets on Bitcoin Layer 1 such as those issued by the Arch network and RGB++ mapped assets will be supported in the future) in the platform to borrow synthetic assets, $bitUSD.
- A liquidity pool for the BTC/BitUSD trading pair is built within the swap of Unisat and Magic Eden, allowing borrowers to exchange the synthetic assets $bitUSD for BTC and LPs to earn fees from borrowers' exchanges.
- During repayment, borrowers need to return bitUSD to the protocol, thus creating demand for exchanging BTC for bitUSD.



Source: *https://shellfinance.gitbook.io/shell*

- BitUSD is also the asset being liquidated in the liquidation, and anyone can participate in liquidating a position that is pending liquidation. When a vault is liquidated, the liquidator needs to pay off the debt and retrieve the corresponding collateral. The price spread between the collateral and its market value is the liquidator's profit. For instance, in a loan with 30 $ORDI as the collateral and 600 $$BitUSD as the loan asset, the position liquidation primarily follows the following process:

1. When the price drops below $28.5, and the LTV falls below 80%, the position reaches the liquidation level and enters liquidation;
2. A 48-hour Dutch auction for the position will be initiated for the collateral currently valued at $855. Bidders have to provide $BitUSD to acquire the asset pending liquidation, with the starting price at 855 BitUSD and the ending price at 600 BitUSD. The price decreases linearly over time.
3. When executing liquidation through a Dutch auction, the liquidator inputs 700 BitUSD based on the auction price. After Shell Finance deducts the 600 BitUSD debt that needs to be repaid, the remaining 100 BitUSD will be added to the insurance fund.
4. After verifying the liquidator's transaction information, Shell Finance adds the collateral to the PSBT, and the liquidator can receive the 30 ORDI collateral.
5. Shell Finance then uses the oracle to reveal the "secret", which completes the participants' (lender and protocol) signatures so as to transfer the collateral from the vault to the liquidator's address. The price oracle will automatically end the corresponding DLC process.



Source: *https://shellfinance.gitbook.io/shell*

At the same time, we can find that Shell Finance is capable of batch borrowing with an APY of only 10% and can support loans with longer terms.

As mentioned earlier, Shell Finance is also using MuSig2 to create Bitcoin-based LST, using LST assets as new collateral and giving BitUSD to stakers. This further expands the utility of BitUSD and raises the project's upper limit.

UTXO-Based BTCFI Scaling Solutions About to Go Live

The Bitcoin community generally believes that an EVM-compatible BTC Layer 2 has limited innovativeness and potential. However, to explore more complex BTCFI applications, stronger Bitcoin contracts are required. Therefore, many Bitcoin developers have introduced native UTXO-based scaling solutions to innovate BTCFI modes based on the UTXO model. We classify these scaling solutions based on whether they settle on the Bitcoin mainnet:

- Solutions that settle on the Bitcoin mainnet can leverage the mainnet's liquidity and are directly compatible with assets like Runes without cross-chain operations.
- Solutions that do not settle on the Bitcoin mainnet require cross-chain asset deposits.

BTCFI Scaling Solutions for Bitcoin Settlement on the Mainnet

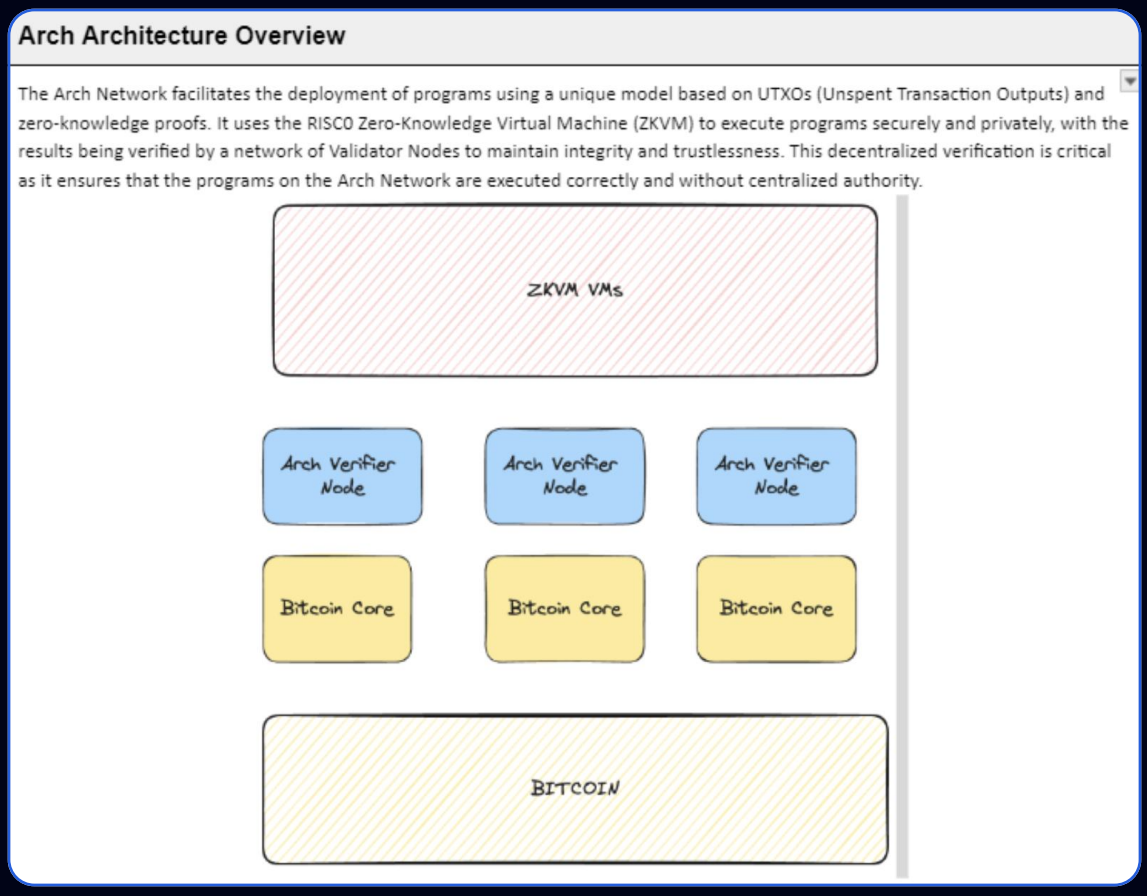
Arch Network: Focused on enhancing computational capacity, it builds a smart contract network using an off-chain zkVM.

Arch leverages a decentralized network of verifier nodes and a specifically built Turing-complete zero-knowledge virtual machine (zkVM) outside the Bitcoin mainnet that can integrate with the Bitcoin mainnet. **This allows it to share liquidity with the Bitcoin mainnet and be compatible with indexers to integrate asset protocols like Runes:**

- **zkVM:** After each smart contract execution, the Arch zkVM generates a ZK proof, which is used to verify the correctness of the contract and state changes.
- **Decentralized Network:** The generated ZK proofs are then verified by Arch's decentralized network of verifier nodes. This network plays a critical role in maintaining the platform's integrity and security. By relying on a decentralized architecture, Arch is committed to ensuring that the verification process is not only secure but also resistant to censorship and single points of failure.
- **Integration with Bitcoin Layer 1:** Once the ZK proofs are verified, the network of verifier nodes can sign the unsigned transactions. These transactions, including state updates and asset transfers determined by application logic, are eventually sent back to Bitcoin. This final step completes the execution process, with all transactions and state updates finalized directly on the Bitcoin blockchain.
- **UTXO Model:** Arch's state and assets are encapsulated in UTXOs, with state transitions conducted following a single-use concept. The state data of smart contracts is recorded as state UTXOs, while the original asset data is recorded as Asset UTXOs. Arch ensures that each UTXO can only be spent once, thus offering secure state management.

It is hoped that DeFi applications compatible with Bitcoin mainnet assets (such as lending and decentralized exchanges) can be built on Arch.





Source: *https://arch-network.gitbook.io/arch-documentation/fundamentals/getting-started*

AVM: A Representation of BTCFI Realized Through Indexer Programming

By introducing a sandbox environment with indexers, sandbox parsers (instruction sets), and a global database, AVM provides Atomicals with an advanced execution environment capable of handling smart contracts and dApps.

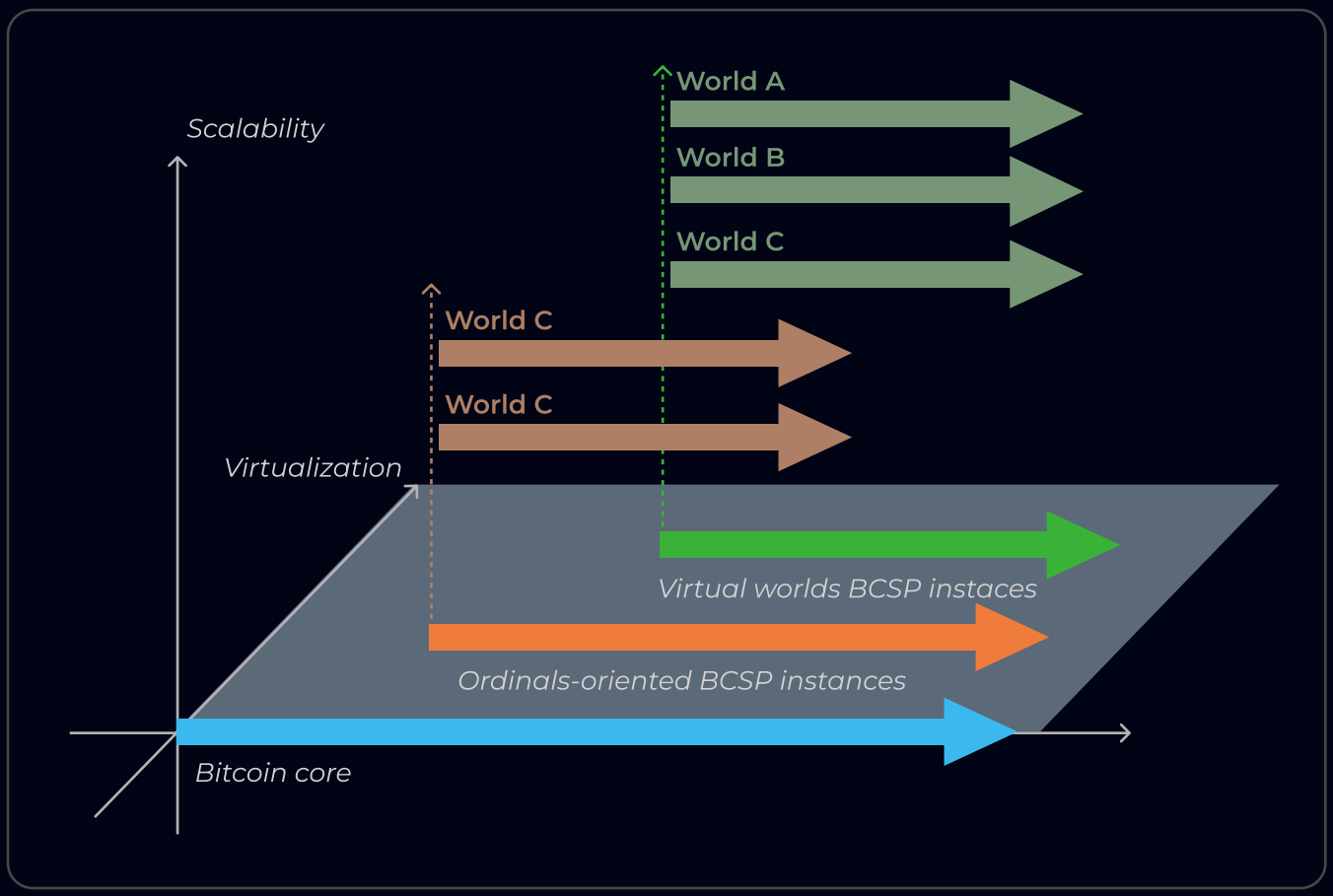
It is equipped with a custom instruction set for performance enhancement, which can also lower gas fees and optimize state transition functions to strengthen parallel processing capabilities, thereby improving throughput and scalability. At the same time, AVM enables interoperability and cross-chain communication.

- In the sandbox operating environment, the entire virtual machine operates in a controlled, isolated environment, ensuring that execution within and outside of the sandbox will not interfere with each other;
- State hashing allows participants to verify whether their indexer's state is correctly synchronized, thus preventing potential attacks due to state inconsistencies.

AVM enables the Atomicals protocol to execute various BTCFI tasks, rather than just the simple token issuance as before.

AVM enables the Atomicals protocol to execute various BTCFI tasks, rather than just the simple token issuance as before.

Fractal Bitcoin: It utilizes the existing Bitcoin architecture to parallelly scale the BTCFI system.

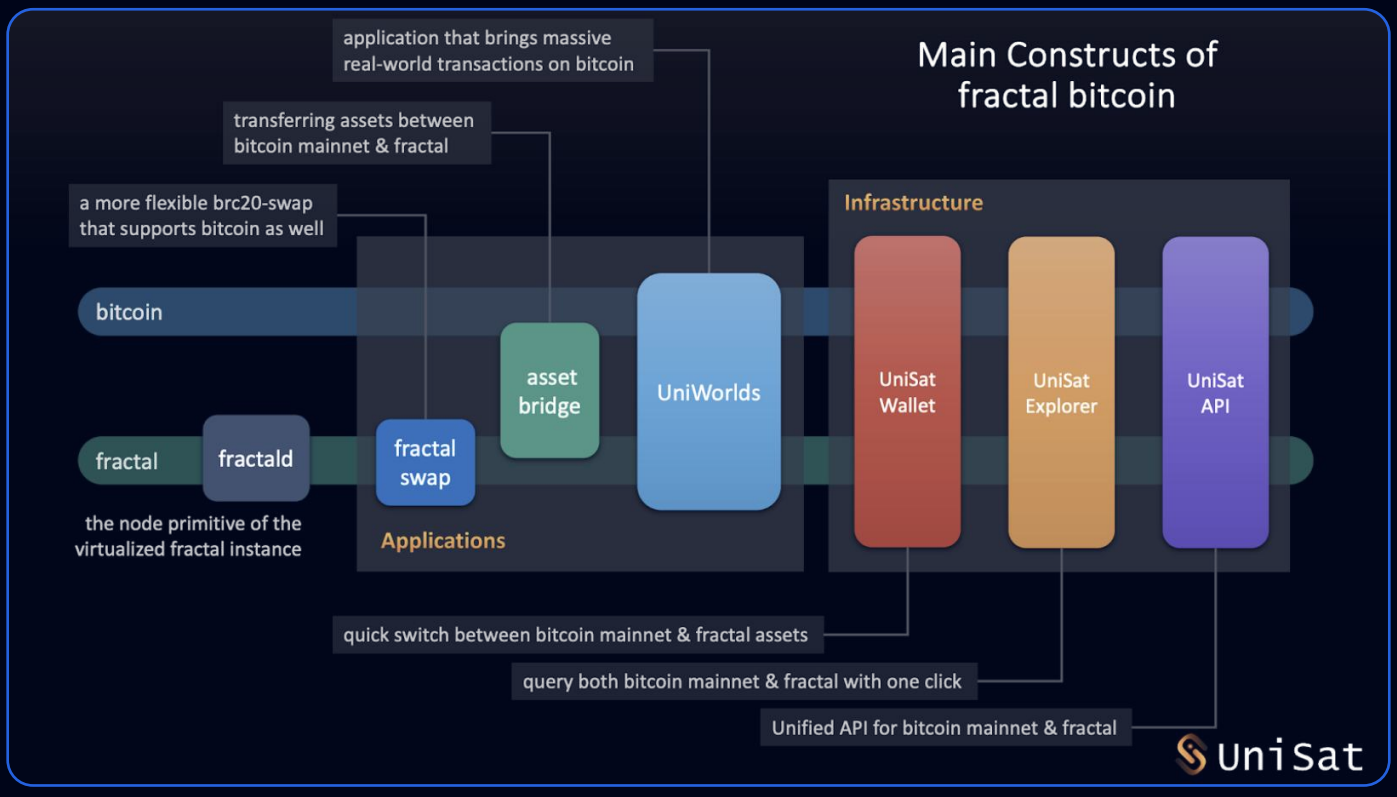


Source: *https://arch-network.gitbook.io/arch-documentation/fundamentals/getting-started*

Fractal Bitcoin is a self-replicating scaling method that encapsulates the entire Bitcoin Core into a deployable and operable blockchain software package called the Bitcoin Core Software Package (BCSP). One or multiple BCSP instances can run independently and are linked to the Bitcoin mainnet through recursive anchoring.

Fractal produces a block every 30 seconds, indicating that it is 20 times faster than the Bitcoin mainnet which produces a block every 10 minutes. It indiscriminately supports and is compatible with all protocols on the mainchain (such as Ordinals and BRC-20) and runs in sync with the mainchain at a different physical settlement rate. The mainnet miners mine a Fractal block every 90 seconds. At the same time, Fractal retains the ability to optionally settle and anchor on the mainnet through inscriptions.

- On the one hand, Fractal aligns relatively well with the mainchain consensus, facilitating protocol-level interoperability.
- On the other hand, Fractal frees itself from the physical constraints and historical burdens of the mainchain and removes some codes that once existed but are no longer practically meaningful. This streamlines the system implementation while retaining full consensus, resulting in a simpler and more lightweight implementation.
- It can implement opcode proposals like OP_CAT faster than the BTC mainnet while following the same basic path as that of Bitcoin upgrade but at a quicker pace. In the future, BTCFI contracts for inscriptions can be realized via scripts.



Source: *https://fractal-bitcoin.notion.site/Fractal-Bitcoin-Public-b71cbe607b8443ff86701d41cccc2958*

Mining Incentive Model

50% of Fractal's tokens are mined, 15% are allocated to ecosystem projects, 5% to investors, 20% to advisors and core contributors, and 10% to partnerships and liquidity. It is evident that its economic model is closely related to miners.

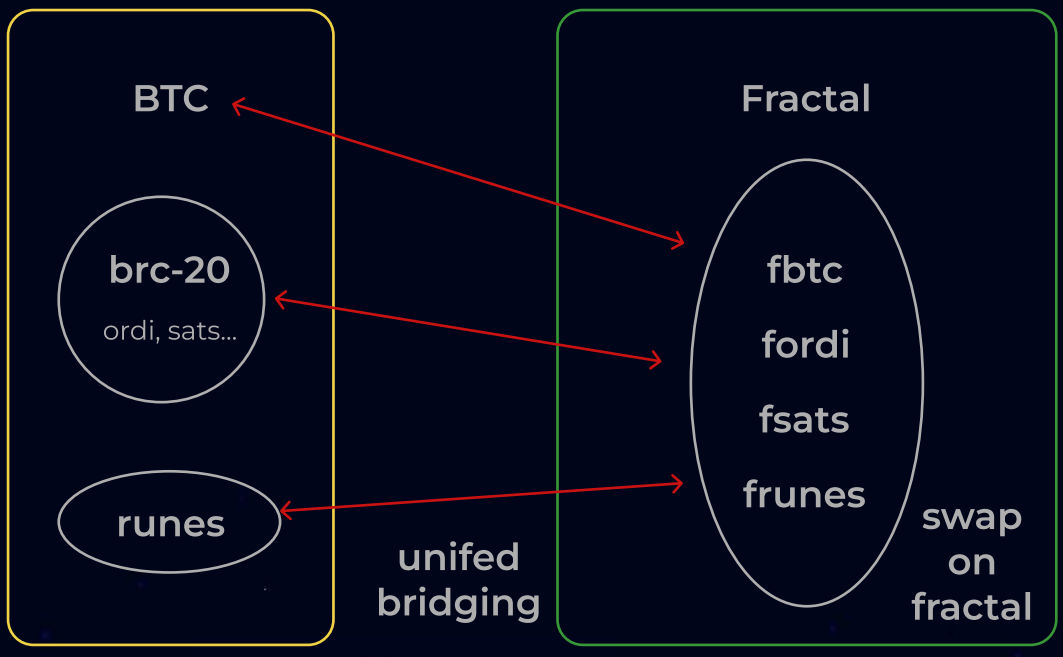
Fractal has innovatively introduced a mining method called "rhythmic mining", where 2/3 of the blocks are freely mined, and 1/3 are collaboratively mined. ASIC miners and pools can use existing mining rigs to mine Fractal while also mining on the Bitcoin mainnet. Fractal Bitcoin's earnings are used to incentivize miners, and its computing power is leveraged to protect the network from potential 51% attacks.

Ecosystem Progress

The Fractal Bitcoin mainnet will launch on September 9. In its ecosystem, there are already multiple NFT projects such as Fractal Punks, honzomomo, Nodino, FractalStone, Fractal Puppets, and MEBS. Additionally, asset issuance platform SatsPump.fun, AMM Pizzaswap, blockchain gaming infrastructure UniWorlds, and NFT generation platform InfinityAI.

Fractal Bitcoin will directly activate OP_CAT upon mainnet launch. The combination of OP_CAT with Fractal's high capacity will make complex Bitcoin applications possible.

Regarding asset migration, BTC and other mainnet assets can also exist on Fractal Bitcoin as wrapped BRC-20 assets.



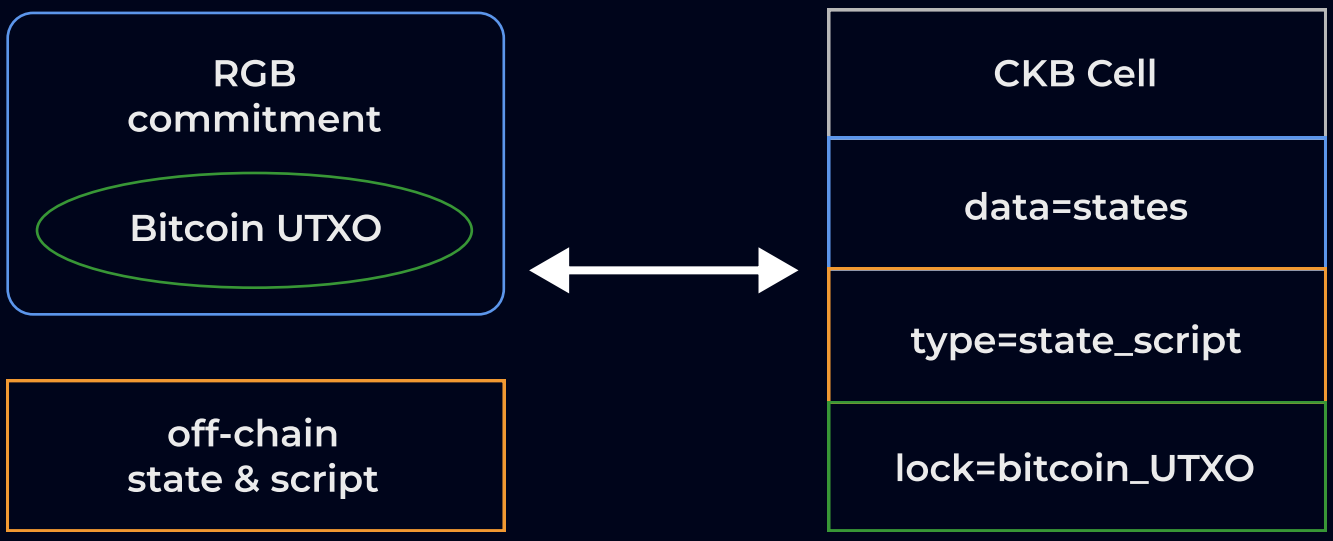
Source: *https://unisat-wallet.medium.com/2024-07-unisat-swap-product-important-update-e974084074a1*

In general, whereas the Bitcoin mainnet focuses on high-value assets, Fractal Bitcoin aims to serve as a storehouse for secondary assets, fostering innovation in assets and applications. However, it remains to be seen whether Fractal Bitcoin will give rise to blue-chip assets and high-quality applications.

RGB++: Developing BTCFI Unique to the UTXO Model

RGB++ utilizes Turing-complete UTXO chains (such as CKB or other chains) as shadow chains to handle off-chain data and smart contracts, thereby further enhancing Bitcoin's programmability.

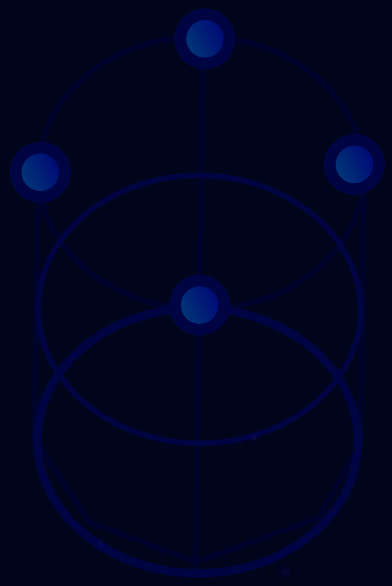
The shadow chain's UTXOs are homomorphically bound to Bitcoin's UTXOs to ensure consistency in state and assets between the two chains and guarantee security. As a result, RGB++ can support Bitcoin mainnet assets like Runes, and its assets can be directly mapped to the Bitcoin mainnet and extended to all Turing-complete UTXO chains.



Source: *https://github.com/ckb-cell/RGBPlusPlus-design/blob/main/docs/light-paper-en.md*

Fully leveraging UTXO model's strengths, RGB++ enables many unique BTCFI features:

- **Leap without bridges through UTXO homomorphic binding:** Assets on RGB++ can move back and forth between the Bitcoin mainnet and one or more L2s without relying on the Lock-Mint paradigm of traditional cross-chain bridges. This approach mitigates many risks associated with traditional cross-chain bridges and offers significant advantages in cross-chain response speed and liquidity aggregation, thus tremendously facilitating the DeFi ecosystem.
- **UTXO transaction model suits intent-driven transaction scenarios well:** A transaction can be completed as long as the desired input and output information is signed and submitted on-chain for verification. For instance, a UTXO input and an output that undertakes asset purchases can be used to place bids for asset transactions without worrying about the intermediate transaction details.
- **UTXOSwap is now live on the mainnet:** The actual experience is almost indistinguishable from Uniswap. UTXOSwap breaks down each transaction into two steps. First, the user submits their intent on-chain. Second, the aggregator aggregates and merges everyone's intent and then initiates a transaction to interact with the liquidity pool.
- **UTXO has a contract script nested mechanism, where a series of continuous transactions can be generated with just one operation, simplifying users' interaction process:** The output from one transaction can be directly used as the input parameter for the next transaction, thus rapidly generating a sequence of interconnected transaction commands.



Conclusion: BTC Has Entered the Mainstream Market, and Its Future Price Surge Will Eventually Drive BTCFI Development

Amid the current downturn in the inscription market and Bitcoin's decline, we might feel pessimistic about BTCFI. However, we must keep in mind that the key distinction from other ecosystems is that Bitcoin will undoubtedly rise in price and keep attracting new retail investors in the future. Bitcoin has been frequently mentioned in this year's U.S. elections.

The U.S. will consider Bitcoin part of the Federal Reserve, and Russia will legalize mining. Mainstream society is actively embracing Bitcoin. At the Bitcoin Conference in Nashville, even moms with kids and every Uber driver are either already Bitcoin holders or preparing to be, treating it as a safe-haven asset.

When Bitcoin reaches new highs, various Bitcoin-denominated assets within its ecosystem will rise correspondingly. This will naturally boost the market demand for BTCFI use cases, such as borrowing with collaterals to purchase more assets or engaging in staking for yield.

In addition, two facts are often overlooked:

- In the past two cycles, Ethereum assets like ICOs and NFTs were dominant. New crypto users, possibly drawn into the market by celebrities releasing NFTs, often chose Ethereum wallets like Metamask. They were accustomed to buying Ethereum for interactions like airdrops and memes and using their idle Ethereum to participate in DeFi.
- In this cycle, Bitcoin has become the dominant one. Users might enter the market after encountering Bitcoin in the U.S. elections or later due to its continuous price surges. They tend to opt for Bitcoin wallets like Unisat and are used to purchasing Bitcoin. They may also use their idle BTC to participate in BTCFI.
- Some still believe that Bitcoin should return to its "digital gold" narrative. However, after the Taproot upgrade and the launch of the ordinals protocol, new retail investors and Bitcoin OGs interested in new use cases for Bitcoin have emerged as a powerful new force. They will be at the forefront of BTCFI innovation, continuously attracting new Bitcoin holders and educating other large Bitcoin holders and miners.

About HTX Ventures

This article is a product of diligent work by the HTX Research Team that is currently under HTX Ventures. HTX Ventures, the global investment division of HTX, integrates investment, incubation, and research to identify the best and brightest teams worldwide. With more than decade-long history as an industry pioneer, HTX Ventures excels at identifying cutting-edge technologies and emerging business models within the sector. To foster growth within the blockchain ecosystem, we provide comprehensive support to projects, including financing, resources, and strategic advice.

HTX Ventures currently backs over 300 projects spanning multiple blockchain sectors, with select high-quality initiatives already trading on the HTX exchange. Furthermore, as one of the most active FOF (Fund of Funds) funds, HTX Ventures invests in 30 top global funds and collaborates with leading blockchain funds such as Polychain, Dragonfly, Bankless, Gitcoin, Figment, Nomad, Animoca, and Hack VC to jointly build a blockchain ecosystem. Visit us here.

Feel free to contact us for investment and collaboration at VC@htx-inc.com



Disclaimer

1. The author of this report and his organization do not have any relationship that affects the objectivity, independence, and fairness of the report with other third parties involved in this report.
2. The information and data cited in this report are from compliance channels. The sources of the information and data are considered reliable by the author, and necessary verifications have been made for their authenticity, accuracy and completeness, but the author makes no guarantee for their authenticity, accuracy or completeness.
3. The content of the report is for reference only, and the facts and opinions in the report do not constitute business, investment and other related recommendations. The author does not assume any responsibility for the losses caused by the use of the contents of this report, unless clearly stipulated by laws and regulations. Readers should not only make business and investment decisions based on this report, nor should they lose their ability to make independent judgments based on this report.
4. The information, opinions and inferences contained in this report only reflect the judgments of the researchers on the date of finalizing this report. In the future, based on industry changes and data and information updates, there is the possibility of updates of opinions and judgments.
5. The copyright of this report is only owned by HTX Ventures. If you need to quote the content of this report, please indicate the source. If you need a large amount of references, please inform in advance (see “About HTX Ventures” for contact information) and use it within the allowed scope. Under no circumstances shall this report be quoted, deleted or modified contrary to the original intent.

References

https://www.btcstudy.org/2023/04/18/interesting-bitcoin-scripts-and-its-use-cases-part-1-introduction/#note6

https://www.btcstudy.org/2023/04/19/interesting-bitcoin-scripts-and-its-use-cases-part-2-multisig/#%E5%A4%9A%E7%AD%BE%E5%90%8D%E8%84%9A%E6%9C%AC%E7%A4%BA%E4%BE%8B

https://www.btcstudy.org/2023/04/21/interesting-bitcoin-scripts-and-its-use-cases-part-3-time-locks/

https://www.btcstudy.org/2022/09/07/on-the-programmability-of-bitcoin-protocol/#%EF%BC%88%E4%B8%89%EF%BC%89%E6%AF%94%E7%89%B9%E5%B8%81%E5%8E%9F%E7%94%9F%E7%9A%84%E5%90%88%E7%BA%A6%E5%BC%8F%E5%8D%8F%E8%AE%AE%E7%9A%84%E7%89%B9%E7%82%B9

https://learnmeabitcoin.com/technical/script/

https://dergigi.com/2022/06/27/the-words-we-use-in-bitcoin/

https://medium.com/summa-technology/bitcoins-time-locks-27e0c362d7a1

https://mp.weixin.qq.com/s/SyZgWBBq1dPkQx8HOAh60w

https://arch-network.gitbook.io/arch-documentation/fundamentals/getting-started

https://learnblockchain.cn/article/8692

https://shellfinance.gitbook.io/shell

https://www.btcstudy.org/2022/08/15/what-are-partially-signed-bitcoin-transactions-psbts/

https://learnblockchain.cn/article/8754

https://multicoin.capital/2024/05/09/the-dawn-of-bitcoin-programmability/

https://www.theblockbeats.info/news/35385

https://foresightnews.pro/article/detail/61617

https://learnblockchain.cn/article/8094

https://www.panewslab.com/zh/articledetails/n36m8a636d3w.html

https://blog.csdn.net/mutourend/article/details/135589245

https://liquidium.fi/

https://foresightnews.pro/article/detail/39035

https://blog.csdn.net/dokyer/article/details/137135135

https://fractal-bitcoin.notion.site/2024-03-Fractal-Bitcoin-FAQ-CN-2faefea58cd04885a376920fce92e632?pvs=25

https://www.panewslab.com/zh/articledetails/iv4jgb5c.html