In light of the recent popularity of the BRC-20 protocol, Goshen's dev team proposes a transformative solution inspired by both Ordinals and BRC-20. This initiative aims to add an Ethereum Virtual Machine (EVM) execution layer to Bitcoin, boasting several key advantages:

1. Leverages Bitcoin's SigWit mechanism, reducing dApp transaction fees compared to Ethereum.**
2. Unlike the Ordinals protocol, which necessitates two transactions(commit and reveal) for data upload, Goshen's protocol requires just one, simplifying user and toolchain construction.**
3. The assets deployed in the EVM can be efficiently controlled, eliminating the need for split transactions like BRC-20 tokens even for basic transfers.**
4. Offers full compatibility with the Ethereum ecosystem, eliminating the need for off-chain toolchain redevelopment.

Motivation

Despite having a built-in scripting language, Bitcoin is somewhat limited in terms of functionality, especially when compared to other smart contract platforms like Ethereum. This restricts the scope of applications that can be developed on the Bitcoin network. The Segregated Witness (SegWit) protocol upgrade, however, has improved Bitcoin's capacity and significantly reduced transaction fees. Therefore, the desire to create an EVM execution application layer for Bitcoin was born. The goal is to allow users to invoke EVM smart contracts directly in the Bitcoin network.

Design

The Segregated Witness protocol is implemented to reduce transaction fees. It allows users to send Bitcoin transactions from a unique P2WSH address, with the EVM invoking transaction data encoded in the witness data. The distinctive redeem script first discards the invoke data from the witness data stack before behaving like a regular redeem script. In contrast to the Ordinals protocol, which encodes data directly in the script and requires one commit and one reveal transaction, our protocol ensures P2WSH address stability and data reveal in one transaction, enhancing user-friendliness.

Implementation

Our solution includes different elements: a specific format for EVM data, an EVM enabled P2WSH (version 0 pay-to-witness-script-hash), and the execution of the EVM. Unlike the Ordinals protocol, our EVM P2WSH is stable, revealing the data directly in a single transaction. The EVM execution does not require an additional fee, and to prevent DDOS attacks, the execution gas is capped at 10,000,000.

Discussion

The rationale behind this protocol is its design to be orthogonal to other aspects of the Bitcoin protocol. It can be used with other layer one and layer applications without any modification to blocks, transactions, or network protocols. Hence, it can be adopted immediately or disregarded, without impacting current users.

Backward compatibility

The Goshen protocol is entirely backward-compatible and doesn't necessitate any changes to the Bitcoin network. It offers an innovative solution to the Bitcoin community, providing the benefits of Ethereum's EVM to Bitcoin users. With this step, we believe we can unlock the next level of growth and development for the Bitcoin network and its users.

Specification

This new layer will utilize two types of EVM invoke actions: EVM call and EVM deploy.

EVM Address

The redeem script hash is 32 bytes, while the EVM address is only 20 bytes. To maintain compatibility with the existing Ethereum ecosystem, we derive the EVM address by taking the ripemd160 of the script hash.

EVM Data Format

In the EVM Call and Deploy, we specify the From Address, To Address, and the Data. The call and deployment actions are then transformed into witness data, which is encoded accordingly. This is a crucial step as it ensures that the size of the encoded data stays within the allowable limits, thus preserving the integrity and efficiency of the transactions.

EVM P2WSH

The redeem script for EVM has a set of DROP opcodes appended to it, making it an extended version of the regular redeem script. The examples provided demonstrate an EVM enabled version 0 pay-to-witness-script-hash (P2WSH), as well as an EVM enabled 1-of-2 multi-signature version 0 P2WSH.

EVM Execution

In an effort to prevent any potential DDOS attacks, the EVM execution does not require any extra fees. However, the execution gas is limited to 10,000,000.

Terminology and Notation

This solution introduces some new terms and concepts related to Bitcoin's EVM layer, including EVM addresses, EVM data formats, EVM P2WSH, and EVM execution. These represent essential components of this EVM execution layer and each serves a specific function within the system.

Conclusion

Goshen's Layer2 solution for Bitcoin, an EVM execution layer, represents a paradigm shift for Bitcoin's development and offers users more functionalities and possibilities. Inspired by both Ordinals and the BRC-20 protocol, it leverages the strengths of Bitcoin's SegWit mechanism and the Ethereum ecosystem.

This protocol is designed to be fully compatible with the current Bitcoin network, and users can choose to adopt it immediately without any impact on their existing operations. The Goshen team is confident that this innovative solution will bring Bitcoin to new heights, further extending the capabilities of its scripting language and making it more efficient and user-friendly for developing applications. With this significant stride, we are indeed excited for the future of Bitcoin and its continuous evolution.

We've publish our proposal at Github:bevm/bip.mediawiki at 75687cb7a447be181c12e5bb23912251082bec19 · goshennetwork/bevm · GitHub, feel free to comment.