

Bitcoin Virtual Machine

Hyperscale Bitcoin indefinitely

<https://bvm.network>

Jan 5, 2024

1. Introduction

James A. Donald jamesd@echeque.com

Sun Nov 2 18:46:23 EST 2008

Satoshi Nakamoto wrote:

> I've been working on a new electronic cash system that's fully
> peer-to-peer, with no trusted third party.
>
> The paper is available at:
> <http://www.bitcoin.org/bitcoin.pdf>

We very, very much need such a system, but the way I understand your proposal, it does not seem to scale to the required size.

Figure 1. The very first response to the Bitcoin whitepaper

Bitcoin inadequacies on the decentralized web are showing. This is largely because most web applications cannot run on Bitcoin for two reasons:

1. Lack of smart contracts
2. Scalability constraints.

To solve these two issues, we propose a new blockchain architecture: Bitcoin Virtual Machine (BVM).

To solve the smart contract problem, we built a Turing-complete virtual machine for Bitcoin. To solve the scalability problem, we implemented horizontal scaling using an infinitely expanding number of Bitcoin L2 blockchains, known as BVM Chains.

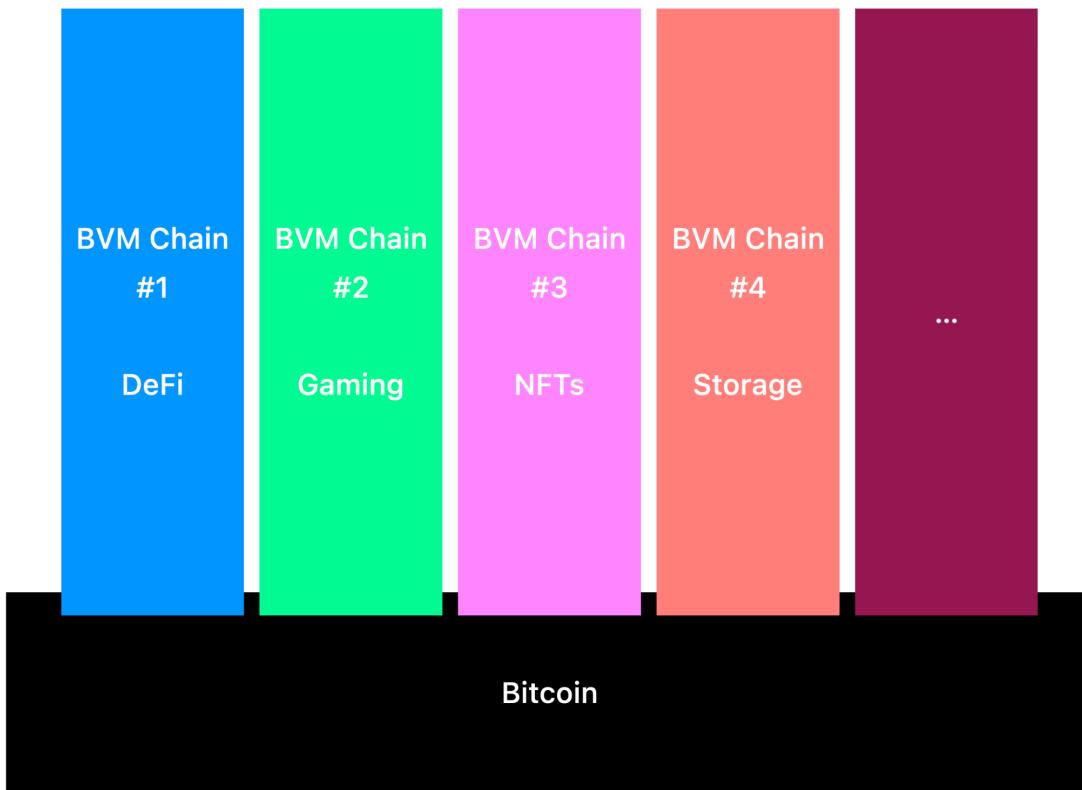


Figure 2. Bitcoin as the blockchain of blockchains

If we execute this well, Bitcoin will become the blockchain of blockchains – with each blockchain optimized for a specific purpose, whether DeFi, Gaming, Payments, or NFTs. All of these blockchains are backed by Bitcoin security.

2. Smart Contracts on Bitcoin

The proposed solution begins with a Turing-complete virtual machine for Bitcoin, known as the Bitcoin Virtual Machine (BVM). BVM works by embedding smart contract transactions into Bitcoin—enabling developers to write smart contracts and build decentralized applications for Bitcoin.

Architecture

BVM is a state machine similar to Ethereum-VM that utilizes Bitcoin as a data layer to achieve transaction-level consensus. This approach allows BVM to function as a general-purpose state machine utilizing Bitcoin's security and data availability without additional network or consensus protocol.

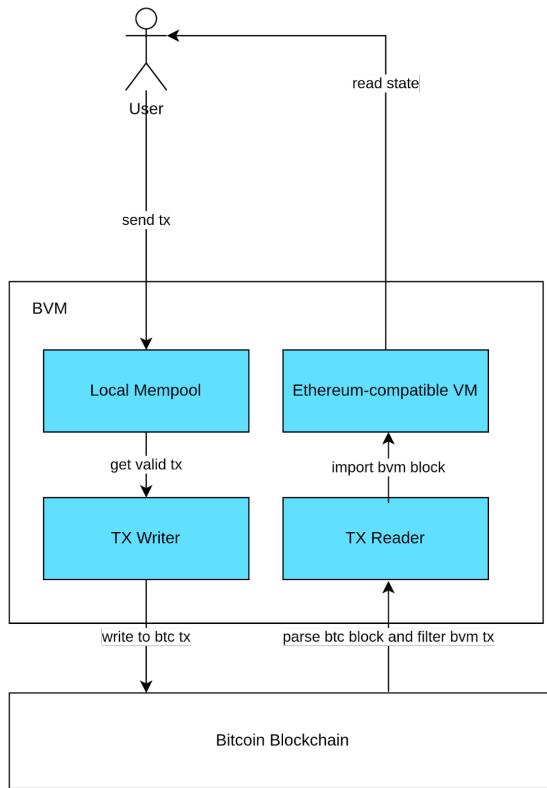


Figure 3. Bitcoin Virtual Machine Architecture

Local Mempool

As part of the system, each node maintains a collection of transactions that the user sends to the node in a local mempool. The mempool helps to verify the validation of the transaction before it is written to Bitcoin.

This mempool is local because it is not shared with other nodes. The transaction is only exposed publicly when written to a Bitcoin transaction (the process of TxWriter) and broadcast to Bitcoin.

TxWriter

TxWriter lets us embed BVM transactions into Bitcoin transactions. To do this, we use a technique akin to that of Ordinals to inscribe data into the Bitcoin network.

Essentially, the BVM transaction data is embedded into a Bitcoin transaction via the witness data field. This embedding is done in a way that does not affect the verification process and has no impact on the transaction logic.

The witness data field is only exposed as an unlock script for the previous output. Pushing data to a Bitcoin transaction requires us to conduct a two-phase transaction process. The first phase is the commit transaction, which involves creating an unspent transaction output (UTXO) that indicates the witness/script hash for spending. In the second phase, the revealing transaction exposes the witness data/script containing the BVM transaction.

TxReader

This module is responsible for filtering BVM transactions in every new Bitcoin block and ensuring that the state of the BVM is consistent across all BVM nodes in the network, even in the event of a reorg.

For each Bitcoin block, the BVM will filter the BVM transaction, sort them by gas fees, and then import them to the VM as a BVM block. As Bitcoin has an immutable, deterministic order, every BVM node that runs an honest codebase will have the same state.

Bitcoin reorg is a process that occurs when two or more miners simultaneously add different blocks to the blockchain. This can cause a fork in the chain, where different parts of the network have different versions of the blockchain. TxReader is designed to handle Bitcoin reorgs by reverting the BVM state to the point of the forked branch and then re-importing the valid blocks into the reverted state.

3. A Network of BVM Chains

By bringing smart contracts to Bitcoin, BVM lets developers create DEX, DAO, tokens, data storage, and many other cases on Bitcoin. However, the biggest challenge is Bitcoin's 10-min block time.

We propose a solution to scale Bitcoin without sacrificing decentralization or security. Increasing Bitcoin capability in terms of speed is fundamental to the mass adoption of decentralized applications on Bitcoin.

BVM Chain Architecture

BVM Chains reuse the battle-tested [Optimism](#) codebase. It is a modified version of the OP Stack that adds support for Bitcoin.

Like Optimism, BVM Chains use [Optimistic Rollup](#), a fancy way of describing a blockchain that piggybacks off the security of another blockchain.

In this case, BVM Chains take advantage of Bitcoin's consensus mechanism instead of its own. This is possible thanks to the Bitcoin Virtual Machine, which brings smart contract capability to Bitcoin.

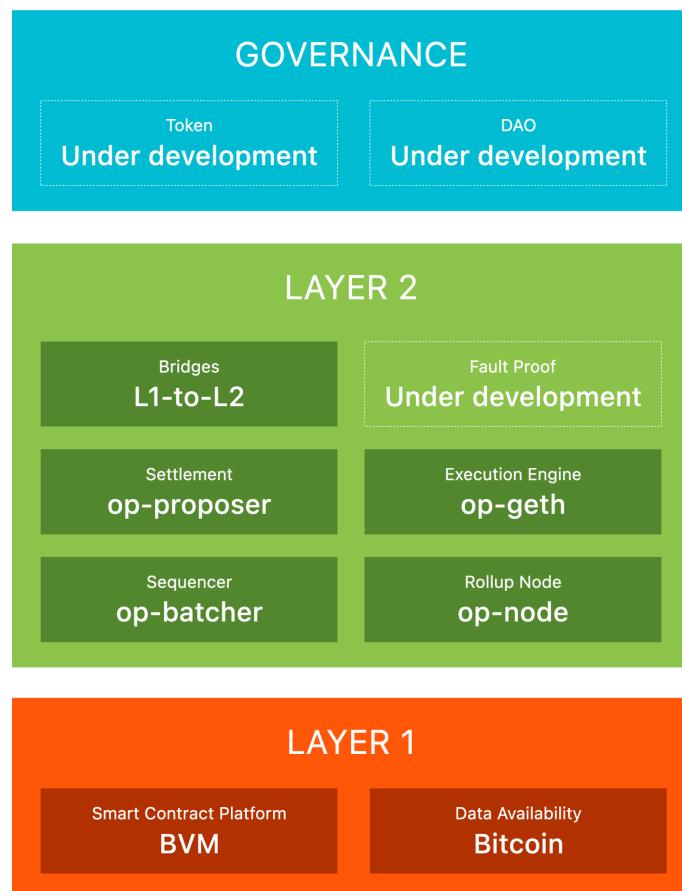


Figure 4. BVM Chain Software Stack

LAYER 1

Data Availability

The foundation of the BVM Chain software stack is the Data Availability component. This component significantly impacts the security model of the entire stack. Everything else in the stack is taken from the Data Availability component.

You can store everything on Bitcoin or use a hybrid storage solution to save costs. For example, you can store the data hash on Bitcoin and the data on another network like Polygon, Celestia, or even IPFS. In this case, Bitcoin is the Data Validation component, and the other networks are the Data Availability components.

Implementation: Bitcoin, Celestia, Polygon, Eigen

Smart Contract Platform

Bitcoin Virtual Machine is a modified version of the EVM that adds smart contracts to Bitcoin. It is used to glue all the components of the BVM Chain software stack together.

Implementation: Bitcoin Virtual Machine

LAYER 2

Sequencer

A sequencer (block producer) determines how transactions are collected and published to Layer 1. The sequencer compresses transactions into batches and writes these batches to the Data Availability component. Then, it writes the data hash to the Data Validation component to ensure data availability and integrity.

Implementation: [op-batcher](#)

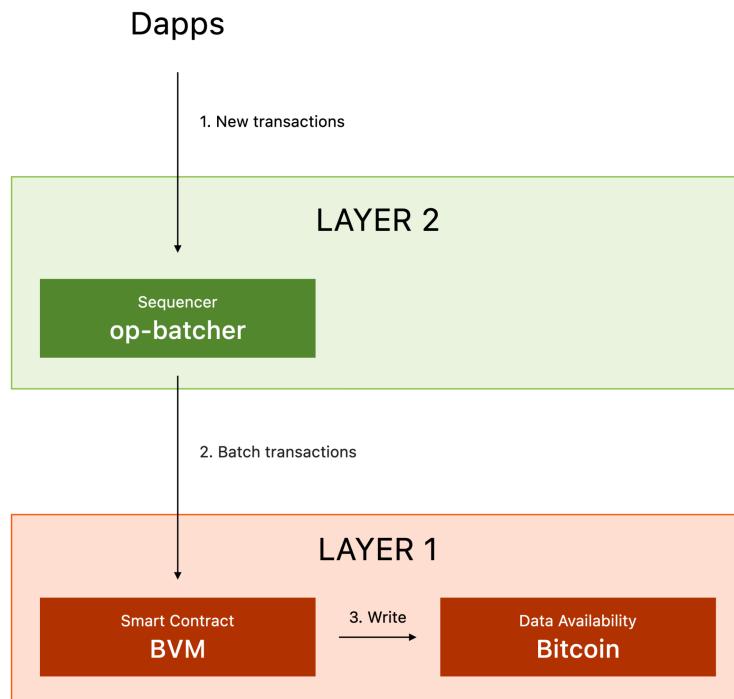


Figure 5. The transaction flow of writing the transactions to the L1

Rollup Node

The Rollup node defines how the raw data stored in the Data Availability component is processed to form the inputs for the Execution Engine. This is also called the Derivation component, which derives the L2 blocks from the L1 blocks.

Implementation: [op-node](#)

Execution Engine

The Execution Engine defines the state transition function. It takes inputs from the Rollup node, executes on the current state, and outputs the new state. op-geth is a modified version of the EVM that adds support for L2 transactions.

Implementation: [op-geth](#)

Settlement

The Settlement component commits the Merkle root of the new state (the output from the Execution Engine) to Bitcoin. Note that the state roots are not immediately valid. In an optimistic rollup setup, state commitments are published onto Bitcoin as pending validity for a

period (currently set at 7 days) and subject to change. The Optimism team is developing a fault-proof process. And once it's completed, we'll add it to the BVM Chain software stack.

Implementation: [op-proposer](#)

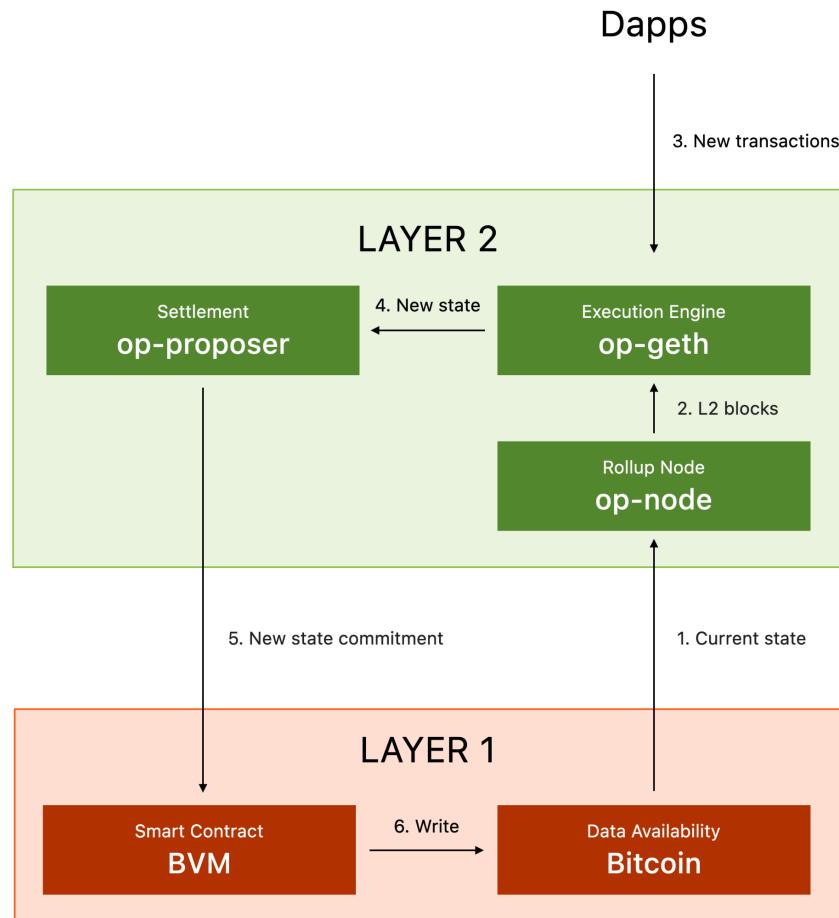


Figure 6. The transaction flow of state processing.

Bridges

Users can deposit Bitcoin assets (BTC and BRC-20 tokens) from Bitcoin to BVM Chains and, vice versa, with the ability to withdraw them.

For example, to deposit BTC from Bitcoin to BVM Chains:

1. Wrap BTC to WBTC via Trustless Bridge
2. Deposit WBTC to BVM Chains via the [L1StandardBridge](#) contract

To withdraw BTC from BVM Chains to Bitcoin:

1. Initiate a withdrawal on BVM Chains
2. Wait for the next Merkle root to be committed to Bitcoin
3. Submit the withdrawal proof
4. After the fault challenge period ends, withdraw your WBTC
5. Unwrap WBTC to BTC via Trustless Bridge

Implementation: [Trustless Bridge](#), [L1StandardBridge](#)

GOVERNANCE LAYER

Token

Each BVM Chain has a governance token for decentralized decision-making.

Implementation: Under development

DAO

Each BVM Chain has its own governance for making network configurations, upgrades, design decisions, development grants, etc.

Implementation: Under development

4. Deploy BVM Chain in a Few Clicks

To make this seamless for developers, we build a fully managed solution to deploy a BVM Chain in just a few clicks.

Developers can choose the best configuration for their applications—whether it's low latency for a game, high throughput for a DeFi application, or low storage cost for an NFT minting app.

By running their own BVM Chain, developers earn sequencer fees, have dedicated throughput for their apps, offer low transaction fees to their users, and have complete control over gas fees, gas block limits, and withdrawal periods.

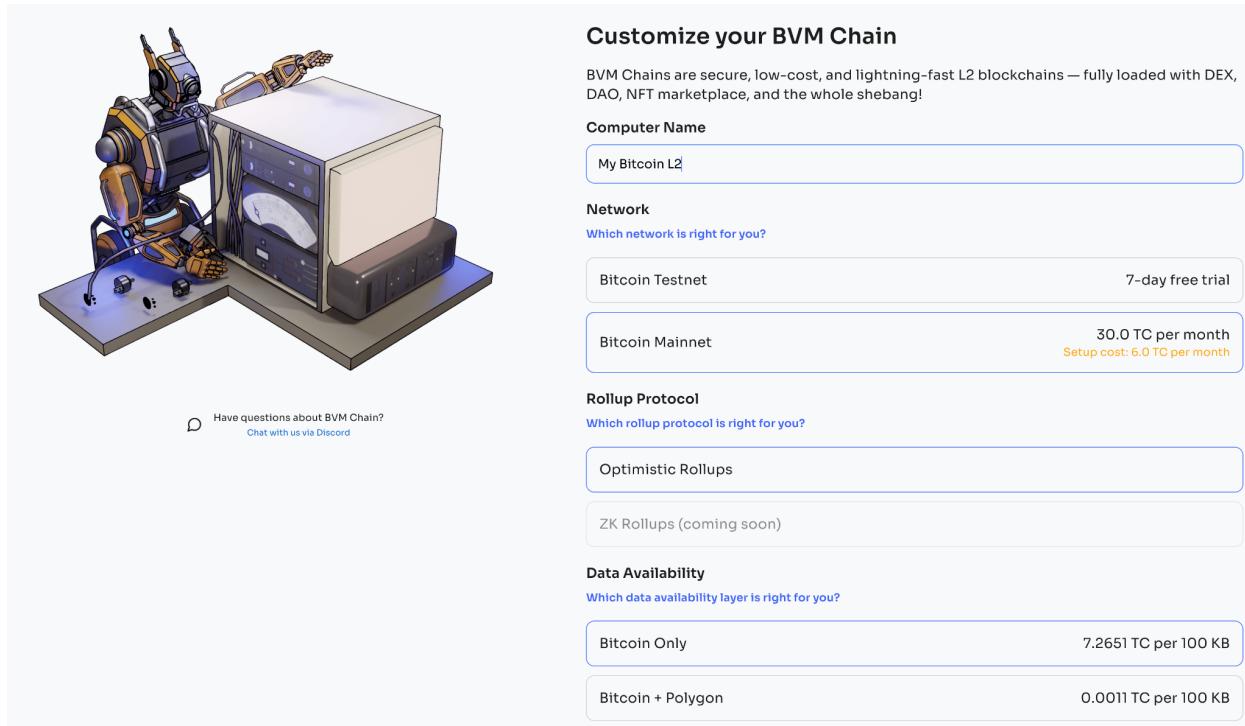


Figure 7. Deploy your own Bitcoin L2 in a few clicks

5. Case Studies

BVM opens up new possibilities for Bitcoin.

In general, we think there are three types of applications built on top of Bitcoin:

- 1) Financial applications.
- 2) Semi-financial applications like gaming, where money is involved, but there is also a heavy non-monetary side.
- 3) Non-financial applications, such as consumer apps, which require low transaction fees, low latency, and high throughput.

Alpha

[Alpha](#) is the first SocialFi app on Bitcoin. Unlike other SocialFi apps, Alpha builds a full-stack crypto social platform. It is a combination of a mobile app at the front end, a set of smart contracts in the back end, and a Bitcoin L2 blockchain.

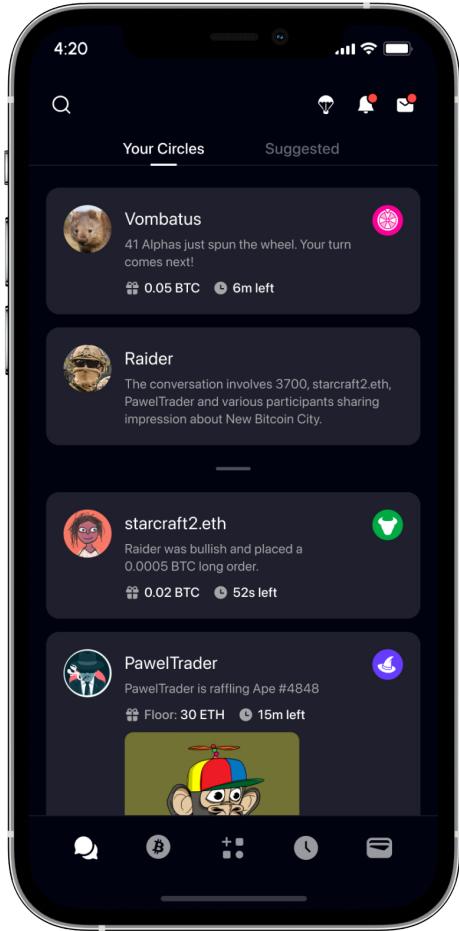


Figure 8. The Alpha superapp

Alpha Chain is the dedicated BVM Chain that powers the Alpha app. In its first 3 months, Alpha Chain handled:

- 23,000+ users
- 877,414 swap transactions
- \$36M (\$838.50 BTC) trading volume on social tokens
- \$3M TVL as the second biggest SocialFi platform in terms of TVL

Satoshi's Gambit

Satoshi's Gambit is the first fully on-chain game on Bitcoin. Note that before BVM, it was impossible to build on-chain games on Bitcoin.

In Satoshi's Gambit, the game states – every move, every game history, and every player's ELO score – is stored on-chain on Bitcoin. The game logic – chess rules, move validation, and the ELO rating system – is entirely programmed in smart contracts powered by BVM.

Satoshi's Gambit was deployed on a BVM Chain Testnet in July 2023, with its deployment on a dedicated BVM Chain mainnet set for early 2024. Satoshi's Gambit BVM Chain will be optimized for low latency, high throughput, and low transaction costs.



Figure 9. The Satoshi's Gambit game

Bitcoin Photos

Bitcoin Photos is the first consumer application on Bitcoin. Much like traditional consumer apps such as Google Photos, you can simply upload your pictures to Bitcoin Photos with a few clicks. Unlike Google, your photos are securely encrypted and stored on Bitcoin – forever.

Bitcoin Photos will be deployed on its dedicated BVM Chain early in 2024. The Bitcoin Photos Chain will also be optimized for low storage costs.

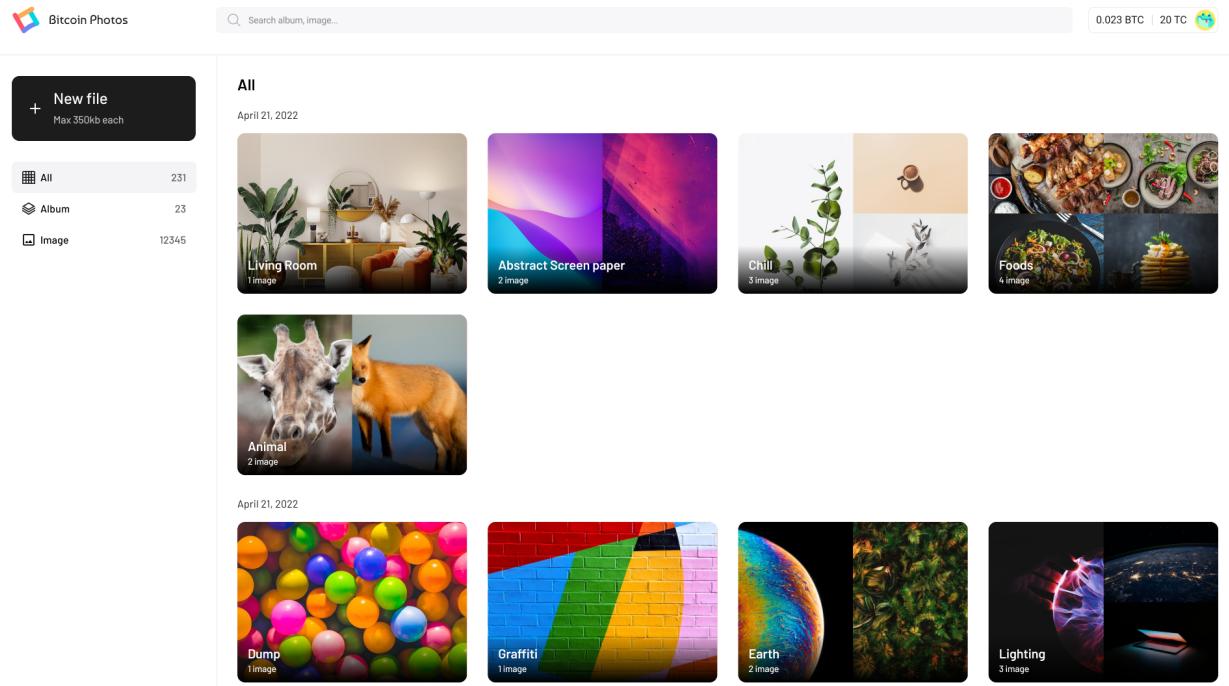


Figure 10. The Bitcoin Photos dapp

6. Conclusion

We have proposed a new blockchain architecture for scaling Bitcoin. To solve the smart contract problem, we built a Turing-complete virtual machine for Bitcoin. To solve the scalability problem, we implemented horizontal scaling using an infinitely expanding number of Bitcoin L2 blockchains, known as BVM Chains. We hope BVM could make Bitcoin as accessible as possible – usable for far more than just a currency. And soon, we'll see DeFi, Gaming, NFTs, Payments, and DAOs, among other uses on Bitcoin.