Dogecoin-Ethereum Bridge

Ismael Bejarano (@CoinFabrik) - Catalina Juarros (@CoinFabrik) - Oscar Guindzberg

Goals

Exchange Dogecoin and an ERC-20 token back and forth in a decentralised manner.

Challenges

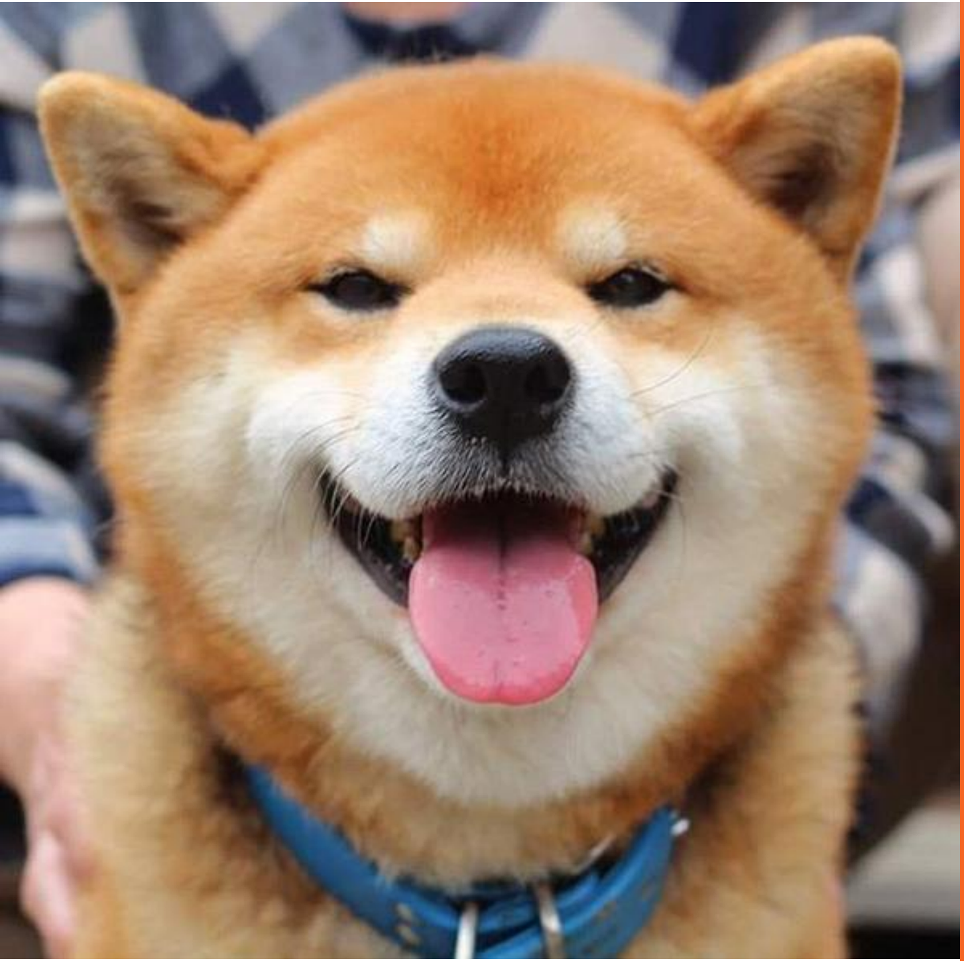
- Keep the same **circulating supply** of Dogecoin.
    - Don’t burn or mint for security reasons.
- Perform the exchange in a **decentralised** way.
    - Exchanges are centralised.
    - Atomic swap requires at least two people.

Existing solutions

- BTCRelay.
  - Only supports one-way transaction verifications.
- RSK Bridge.
  - Supports two-way operations.
  - Controlled by a federation.

More challenges

- Dogecoin uses **Scrypt** as its proof-of-work function.
  - EVM-based verification costs about **100M gas**.
- Storing all the **blocks** is expensive.
  - **200USD** per day, even if the bridge receives no transactions.
- Dogecoin has scripts, but it offers **limited support** for programming.
  - Adding an opcode would require a **fork**.



Solution!

TrueBit

Off-chain Scrypt hash verification using a challenge-response protocol.

TrueBit

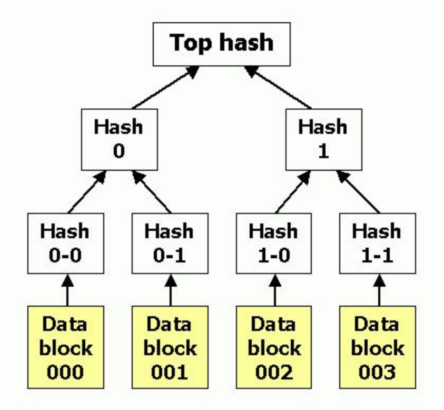
- Scrypt hash is calculated **off-chain**.
- Iterative challenge:
  - Divide the problem into N steps.
  - Binary search to find the first incorrect step.
  - Execute incorrect step in the contract.
- Economic incentives to prevent attacks.
  - Each step must cover the potential response’s cost.

Superblocks

Store the Merkle root of a tree consisting of several blocks.

Superblocks

- Blocks that aren’t relevant for the bridge don’t need to be stored.
- Adds complexity and possible attacks.
- The goal is to disincentivise attacks.



Collateral

Mechanism similar to MakerDAO's stable coin DAI for converting DogeTokens to Dogecoin.

Collateral

- Dogecoin:
    - Doesn’t support complex scripts.
    - New opcode needs a hard fork.
- Collateral:
    - Dogecoin is backed by operators.
    - Operators must deposit ether in order to cover the total amount of Dogecoin in the bridge.
    - Affected by Ether to Dogecoin price fluctuations.

Tools

- Truffle
- Ganache
- Travis CI
- Web3j

Truffle

- Smart contract compilation and deployment.
- Integration tests for Solidity smart contracts and Java agent.
- Unit tests during development.
    - We currently have over 100 unit tests.

Ganache

- Development Ethereum node.
- Automatic mining.
- Infinite balance.

Travis CI

- Test case execution in a clean environment.
- Execute tests on a branch before merging.

Web3j

- Java version of web3.
- Used by ‘agents’ for interacting with the Ethereum blockchain.

Some issues

Some issues

- Truffle
  - Sometimes it doesn’t recompile contracts:
    - Automate compilation and deployment with bash scripts.
    - Remove *build* directory.
    - Force recompilation: *truffle compile --all*.
  - Latest compiler version:
    - Edit dependencies manually.

Some issues

- Ganache
  - Slow for complex contracts.
  - Easy to create transaction collisions.
    - Transactions aren’t signed.
  - A bugged version made Travis CI fail.
    - Hardcode a working version.

Some issues

- No stack trace for debugging.
  - Use error codes instead of *revert*.
  - Use *log0()*, *log1()*, etc. to inspect variable state
  - New *revert with reason* opcode is not yet supported
    - There is still no defined protocol for interpreting *reason*

Some issues

- Possible ‘out of gas’ causes:
  - 32KB per transaction limit.
    - Makes it impossible to deploy very large contracts.
  - Almost any error causes ‘out of gas’ on Ganache.
    - Turn on verbose mode.
    - Test on geth development mode (PoA).

Some issues

- ‘Out of gas’ solutions
  - Separate contracts according to their functionality.
    - Adds complexity and dependencies between contracts.
  - Use libraries.
    - *extern* functions use *delegatecall*.
    - *internal* functions are compiled inline.
    - No access to storage.
  - Use Solidity assembly.
    - Hard to debug.
  - Turn on compiler optimisation.

That’s all!

References

- **Efficiently Bridging EVM Blockchains,** https://blog.gridplus.io/efficiently-bridging-evm-blockchains-8421504e9ced
- **A scalable verification solution for blockchains,** https://people.cs.uchicago.edu/~teutsch/papers/truebit.pdf
- **The Dai Stablecoin System,** https://makerdao.com/whitepaper/DaiDec17WP.pdf
- **Reference implementation of the decentralized Dai Stablecoin issuance system,** https://makerdao.com/purple/