Pwn web3 through Multichain attacks

Workshop

author : six

Agenda

1. Introduction to web3 hacking, concepts / workshop topology
2. Environment setup, system requirements

Scenario 1: Token on two chains, mint using receipt

1. Solidity basics, using remix for compile
2. Exploit visibility, take admin
3. ECDSA Ethereum basics
4. Mint with receipt -> Find the vuln!

Scenario 2: Signature forgery (any chain)

1. Deploy SC on Ethereum chain
2. Compile Substrate with EVM
3. Deploy SC
4. Test ECDSA signature forgery exploit from one to other


```
six ~  > whoami
> Web3 hacker
> Founder of CCTF
> Co-founder of QRUCIAL
> Polkadot Head Ambassador of Eastern Europe
> Just does what he loves.
```

QRUCIAL

Polkadot

CCTF

@DaveTheSix

What the web3 aka...

Reverse engineering buzzwords

& the hype

proof of deposit

bridge

swap

remix ide

crypto

token

DeFi

dApp

decentralized

coin

crypto wallet

bridge

web3

substrate

javascript 😱

metaverse

solidity

blockchain

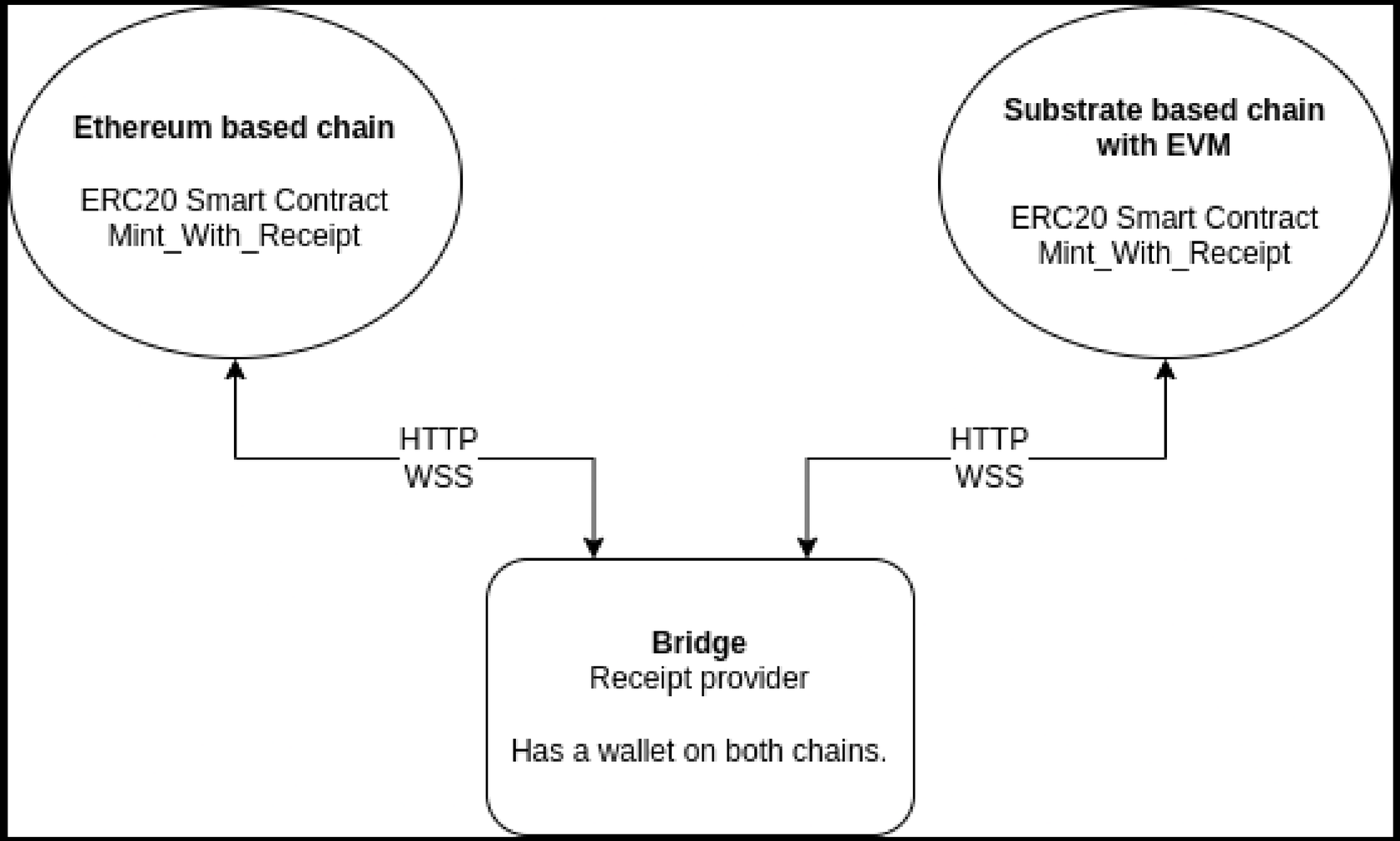
tx

wrapped coin

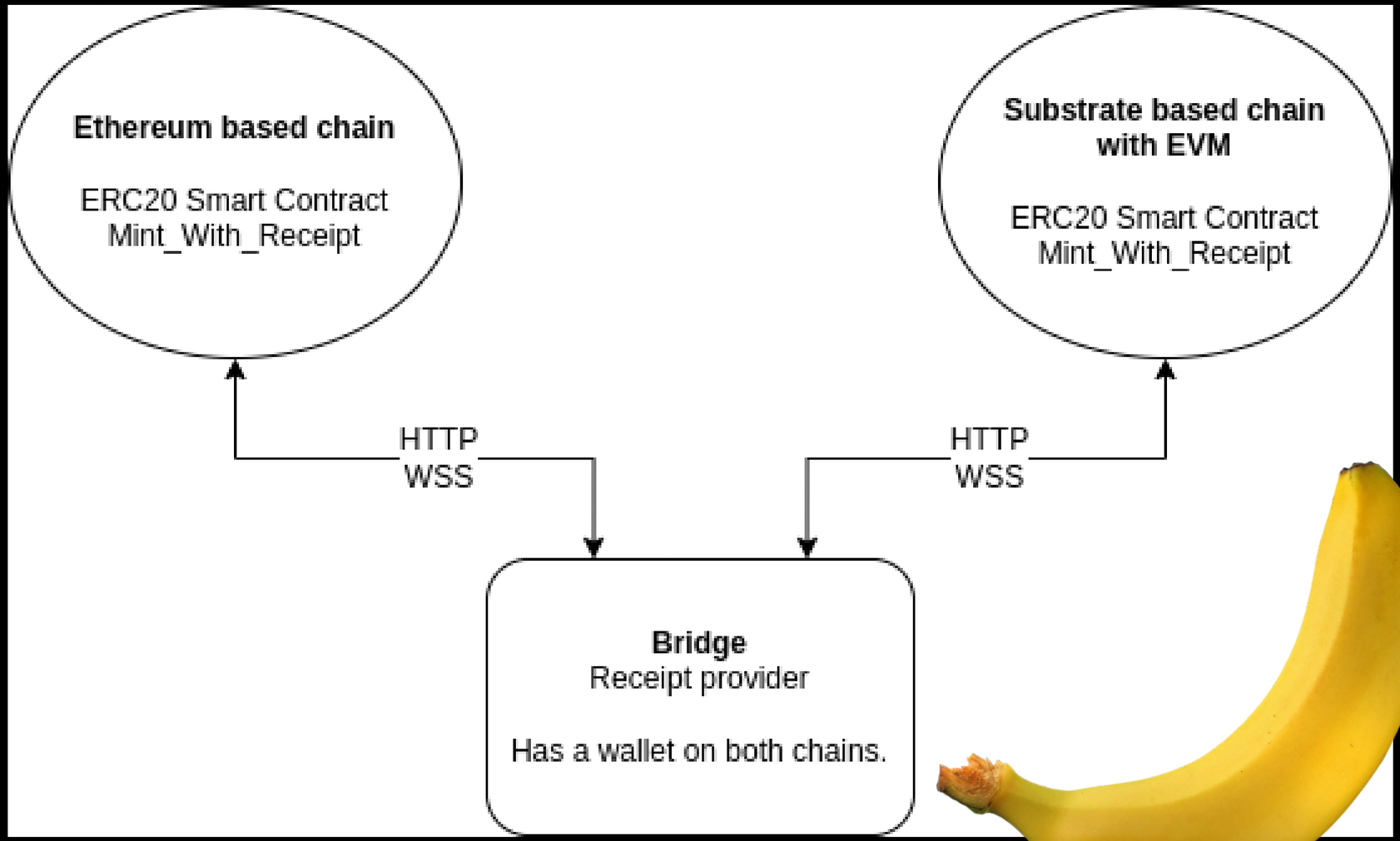
smart contract

signing

Topology


Ethereum based chain
ERC20 Smart Contract
Mint_With_Receipt
Substrate based chain
with EVM
ERC20 Smart Contract
Mint_With_Receipt
HTTP
WSS
HTTP
WSS
Bridge
Receipt provider
Has a wallet on both chains.

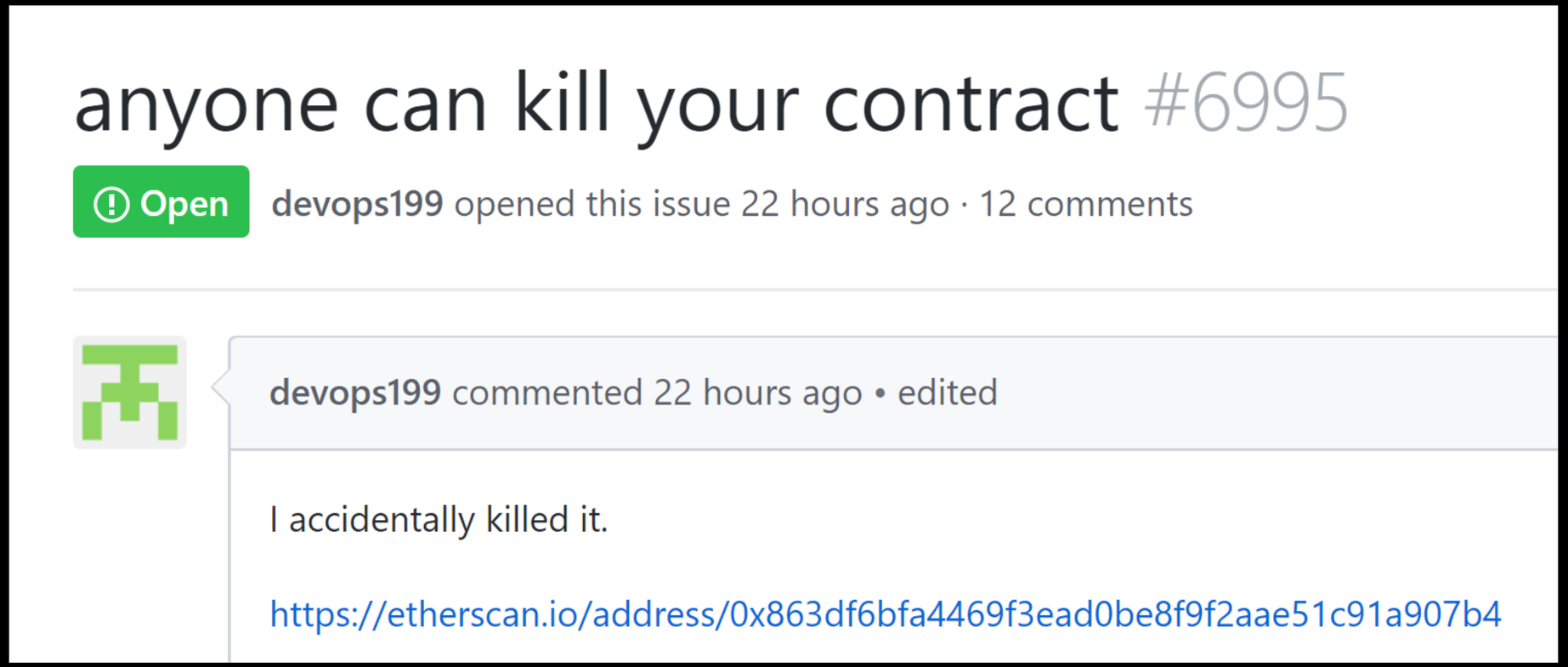
Topology


Ethereum based chain
ERC20 Smart Contract
Mint_With_Receipt
Substrate based chain
with EVM
ERC20 Smart Contract
Mint_With_Receipt
HTTP
WSS
HTTP
WSS
Bridge
Receipt provider
Has a wallet on both chains.

Demo: use wBan bridge live

https://bsc.banano.cc/

Hacking Warmup



Multisig issue: https://github.com/openethereum/parity-ethereum/issues/6995

Practice

- Python3 receipt generator

- Generate receipt

- Exploit the smart contract(s)

TXDATA Replay Attack
or "Receipt Replay Attack"

```
Function: mintWithReceipt(address recipient, uint256 amount, uint256 uuid, uint8 v, bytes32 r, bytes32 s) ***

MethodID: 0x05b084df
[0]:  00000000000000000000070ebc8b2596f023f94c4790df06510265b045f14
[1]:  0000000000000000000000000000000000000008dafa88e340e3ed80000
[2]:  000000000000000000000000000000000000000000000017cefbc69f3
[3]:  000000000000000000000000000000000000000000000000000000001b
[4]:  2aa17615cf385bc09fefbf96968005f05258033704a880407a5ac72e6a395076
[5]:  5347c8812f0bfaa5df77e832f14d618ce535e8900136bb87ec6699dc8c1b6d64
```

wBanano hack TXs:

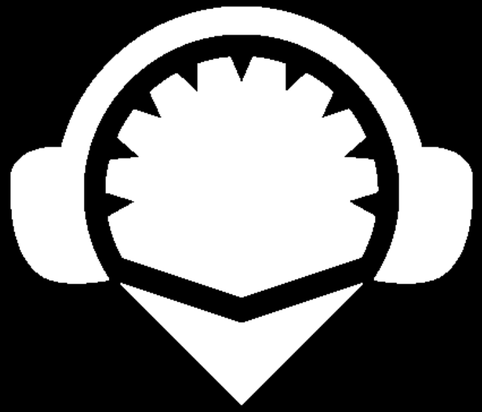
https://polygonscan.com/tx/0xbcf3f1192d63a0d240995619b8896c406d1ba6fa7c2fc81503057d61c98bba41

https://bscscan.com/tx/0x60c3ae26d1a1d2b525a425aacdbde30bf7efdc09a125086cc7aab9b347daf684

Tools to pwn all the Eth

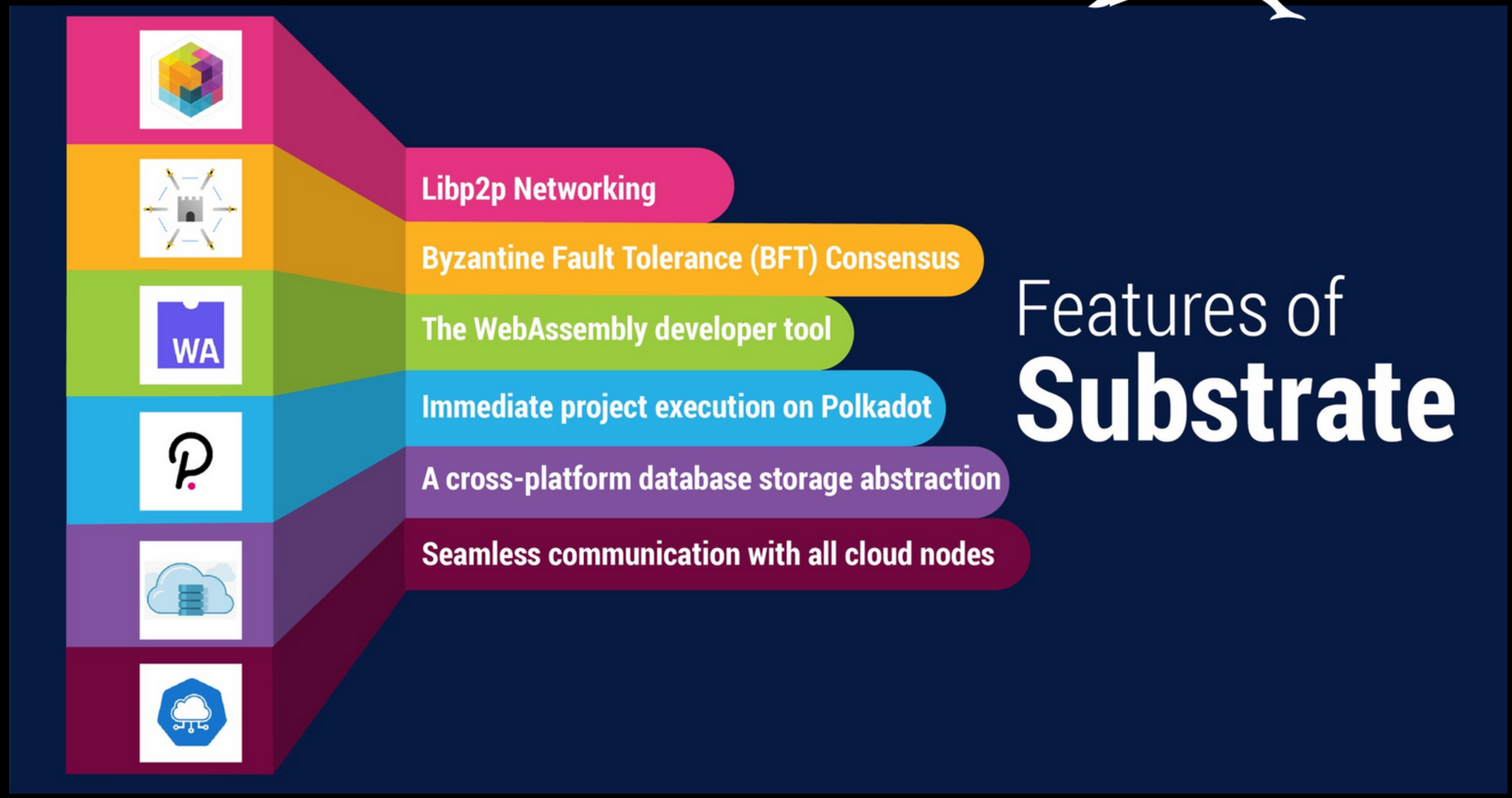
TRUFFLE

Ganache



+ github.com/crytic/awesome-ethereum-security

Hacking Substrate - Why?



Polkadot

Takeaway

Bridges are centralized (not web3!)

Bridges can be web2 hacked

We need interoperability and protocols eg. XCM

zkBridge - https://arxiv.org/abs/2210.00264

We are getting there.

Contact

Matrix: @hexff:matrix.org

Twitter: @DaveTheSix

Telegram: SixTheDave

Git: git.hsbp.org/six

CCTF room: #CCTF:matrix.org

CCTF Telegram: t.me/cryptoctf

Polkadot Hungary: t.me/polkadothungary