Citrea - Bitcoin L2

これは2025年の調査ページ。

2026年以降→ Citrea (2) - Bitcoin L2



https://citrea.xyz/

開発元は Chainway Labs。ケイマン諸島ということになっている。

https://www.blog.citrea.xyz/

https://docs.citrea.xyz/

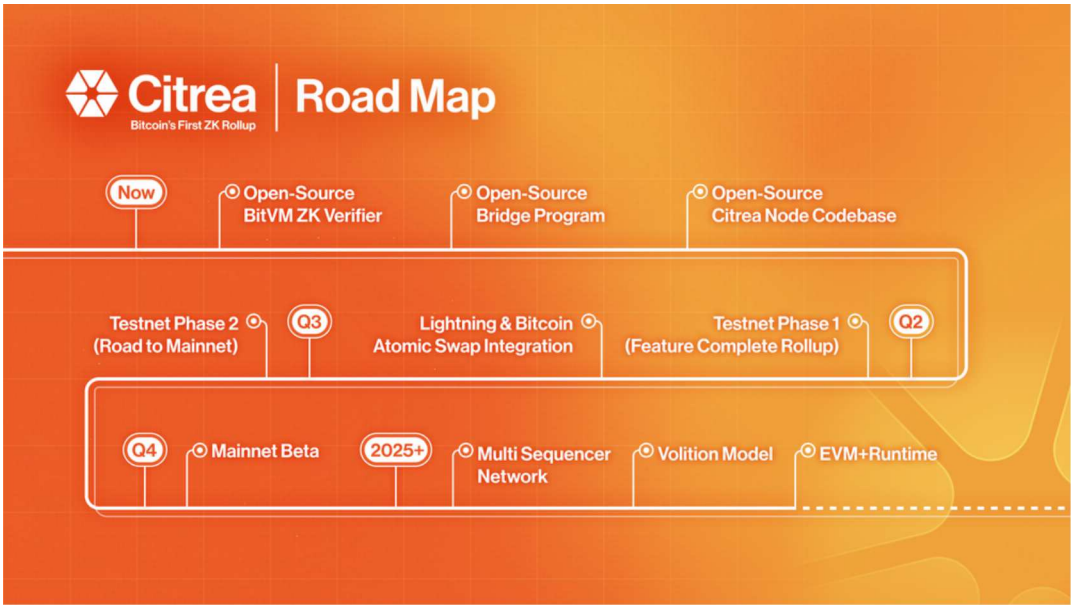
GitHub

https://github.com/chainwayxyz/citrea

https://github.com/chainwayxyz/clementine

Citrea - ChatGPT

ロードマップ(2024年4月)



Technical Specs

Citrea概要 - ChatGPT

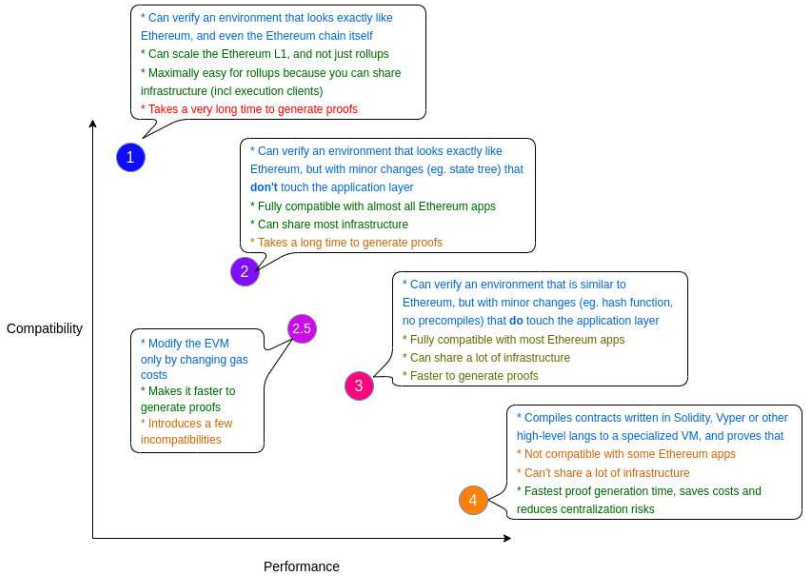
ビットコイン上に構築された初のゼロ知識ロールアップ（ZK Rollup）

Bitcoinを Data Availability layerとして使っている

BitVM2ベースのブリッジ「Clementine」を使用して、ビットコインとCitrea間の信頼最小化された双方向ペグを実現

RISC Zeroを使用してEthereumの仮想マシン（EVM）と完全に互換性のあるType 2.5 zkEVMを構築

Vitalikによる分類で、Type2がバイトコード互換、Type2.5はその最適化あり



zkRollup は zkSNARK、ValidityはzkSTARK？

RISC Zero は zkSTARK系

よくわからないのでGemini Researchに調べてもらった。

ハイブリッドな「証明の証明」システムらしい

まず、L2のトランザクションをバッチ処理した正当性証明のために zkSTARKs を生成する。

それを再帰的にラップ / 圧縮して zkSNARKS を生成する

そういう手法は実際に存在し、そうしているL2もいくつかあるそうだ

L2ステートの検証

zkSNARKでの証明データをBitVMで検証しているそうだ(同じくDeepResearch)

L2 上でも BTC を使う。

cBTC という単位

Citreaトークンみたいなやつはやってないので、そういうのがあったら詐欺注意。

ブリッジ

Clementine

https://www.blog.citrea.xyz/unveiling-clementine/

#64 - Citrea: Scaling Bitcoin's Future with ZK Rollups

v1とv2がある

v1

BitVM2でZKPしている

現在は peg-in は 10BTC の固定額でしか使用できない(固定額になるのはBitVM2 bridgeの仕様)

peg-inしたいユーザが 10BTC を送金

送金先は1つ

「 `N+1` -of- `N+1` のマルチシグ」 or 「200ブロック以上経過したら送金者本人」

HTLC

たぶん `N` 人がverifierなどで `+1` が自分

マルチシグから `N` -of- `N` などに送金するトランザクションを作って自分の分の署名をverifierに送る

verifier `N` 人が署名してトランザクションを展開するのだろう

通常は 200ブロック経過する前に Citrea の管理者たちがマルチシグに署名したトランザクションを展開する

送金先は3つ

`N` -of- `N` のマルチシグ

`OP_RETURN` : Citrea のアカウント

CPFP用のdust

v2

BitVM2方式から Garbled Circle + TOOP方式に変更する予定。

2025/10/06現在で研究中。

https://www.blog.citrea.xyz/clementine-v2-with-garbled-circuits-and-toop/

From Sovereign Rollup to zk-Rollup - Untold Story of Bitcoin Layer 2

testnetでの運用が始まるくらいの2024年8月に書かれた記事。

2025年5月では L1 に ZK証明を書き込んでいる。BitVMを使って Bitcoinスクリプトでの検証ができているかは確認できていない(やり方が分からない)

BitVM1 は2024年3月末が最後のようだが Citrea がどうだったのかはよくわからない。

オフチェーンで検証を行い、オンチェーンには証明を保存するやりかたを Sovereign Rollup

と呼ぶ

このアプローチ自体を「どうなの？」と思っている人々もいるが、クライアント側検証するのを受け入れても良いかと思っている人もいる。

それはよいとして、BitVMを使うといっているプロジェクトについては BitVM 自体がまだ商用レベルにまでなっていない

Citrea が Bitcoin だけで Rollup するようなことを書いているが、この記事の時点では Sovereign Rollup のままである。

BitVM は ZKPのオフチェーン検証でのみ使っている

ブリッジはマルチシグ方式のまま

Citreaはスキャムだということを言いたいわけではなく、他と同じような Rollup 方式なのに「Citreaが最初のRollupだ」と主張しているところを言いすぎだと思っているようだ。

ZKPの検証をBitVMでやれているなら zkRollupということでもよいのでは？

Inscription として ZK証明のデータを上げているが BitVMで検証しているかはぱっと見て分からなかった

Grok

L1に書いたZK証明をBitVMで検証する、と書いているので↑↑から月日が経って方式が変わったのか？

L1で検証するので決済もL1になる。が Citrea は独自のスケジュールを持っていて**1時間**だそうだ。

Nasdaqの記事で「ZK proofsをBitcoin上に1時間ごとに投稿」といっているそうだ。

Clementineブリッジになって設計が変わったことによるものらしいが、ブログではブリッジのファイナリティは表があるものの決済については書いていない。Nasdaqの記事くらいしか情報がないらしい。

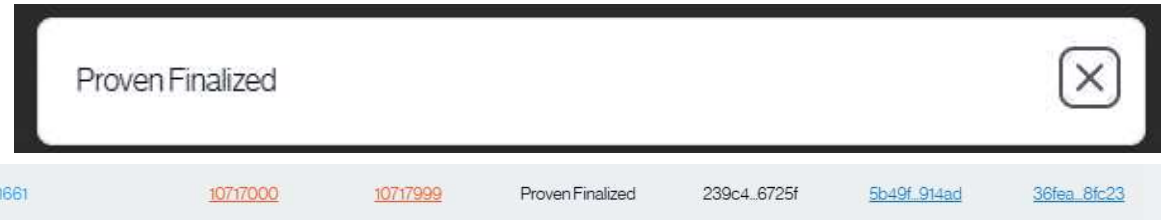
比較表

以下の表で、Bitcoinのブロックタイムと Citreaの決済頻度を比較します。

側面	Bitcoin（L1）	Citrea（L2）
ブロックタイム/投稿頻度	約10分（ブロック生成間隔）	約1時間ごと（ZK proofsの投稿）
トランザクション処理	L1上で直接処理（10分ごと）	L2で継続的に処理、L1で1時間ごと決済
ファイナリティ	ブロックが生成されてから確定（約10分）	ZK proofsがBitcoinに投稿後確定（1時間）

L1で取り込まれたL2ブロックはBatchExplorerで確認できる

testnet4 で送金したトランザクションのブロック 10717054



L2の 2000ブロックをまとめて L1に投げているようだ

commitment:

`239c4ea0496f718590b1048a1c31a44c6602412d9f30eb5774f732a34346725f`

Commitment TXID

シーケンサーがバッチのトランザクション順序を固定するためにBitcoinにハッシュをコミットするトランザクションのID

トランザクションが Inscription 扱い だが見ても分からん。

送金先は key path で、この先も使われるアドレスだった

Proof TXID

バッチのゼロ知識証明と状態差分がBitcoinにinscribedされるトランザクションのIDで、バッチの有効性を検証する

トランザクションが Inscription 扱い だが見ても分からん。

データが大きい

比較
以下の表で、Commitment Transaction IDとProof Transaction IDの違いをまとめます。

側面	Commitment Transaction ID	Proof Transaction ID
目的	トランザクションの順序を固定	バッチの有効性を検証
内容	シーケンサーのハッシュ（Merkle root）	バッチ証明（ZK proofs）と状態差分
タイミング	バッチ処理の初期段階で生成	バッチ証明が生成された後に生成
Bitcoinでの役割	トランザクション順序の固定を保証	バッチの有効性を検証

Inscription になっているのはわかったが、BitVM がどう絡んでくるのだろうか？

testnet4で動かす

testnet4 ノード + citrea フルノード

docker コンテナも用意してある

https://docs.citrea.xyz/user-guide/run-a-node/citrea-testnet/citrea-testnet-executable

`rollup_config.toml` と genesisのデータをダウンロード

実行ファイルは Linux Intel 64bit か Mac Arm 64bit かしかないので、Raspberry Piで動かしたかった私は自力でビルド

試した refs/tags https://github.com/chainwayxyz/citrea/tree/v0.7.1-int-rpc-fix

Raspberry Pi というか Arm 64bit は Mac 以外対応していないようで、ビルド中に

Arm 64bit に対応していないためエラーになった。どれかの Makefile で、サポートしていないプラットフォームとなって終わったと思う。
WSL2で動かすことにした。作業中にしか起動させていないというのもあるが3日経っても同期のような処理が終わらない。こういうログがひたすら出ている(block の番号がたぶん増えている)

log
```
2025-05-15T00:18:46.756032Z  INFO citrea_common::l2: New State Root after l2 block #3196748 is: 0x047ffd60dfbec9a951edadf478596f9b29a71dcf9f5a4238d953df9eab816a1f
```

curlでステータスが取れる。"l2status" が "syncing" なのでまだ同期は終わっていない。

log
```
$ curl -X POST --header "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"citrea_syncStatus","params":[], "id":31}' http://0.0.0.0:8080
{"jsonrpc":"2.0","id":31,"result":{"l1Status":{"synced":"0x1411f"},"l2Status":{"syncing":{"headBlockNumber":"0x9c3bf5","syncedBlockNumber":"0x30df8d"}}}}
```

同期が取れるとこうなる(`jq` で整形)。休み休みしか起動していないとはいえ12日かかった。

log
```
$ curl -s -X POST --header "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"citrea_syncStatus","params":[], "id":31}' http://0.0.0.0:8080 | jq .
{
  "jsonrpc": "2.0",
  "id": 31,
  "result": {
    "l1Status": {
      "synced": "0x14643"
    },
    "l2Status": {
      "synced": "0xa2d612"
    }
  }
}
$ du -h
129G    ./dbs/state
```

```
6.8G    ./dbs/native
2.4G    ./dbs/ledger
138G    ./dbs
138G    .
```

一晩で 2万ブロックくらい進んでいる。

Citrea Faucet

Faucetを受け取るのにローカルで動くフルノードはいらない。。。

Citreaチェーンの通貨単位は cBTC

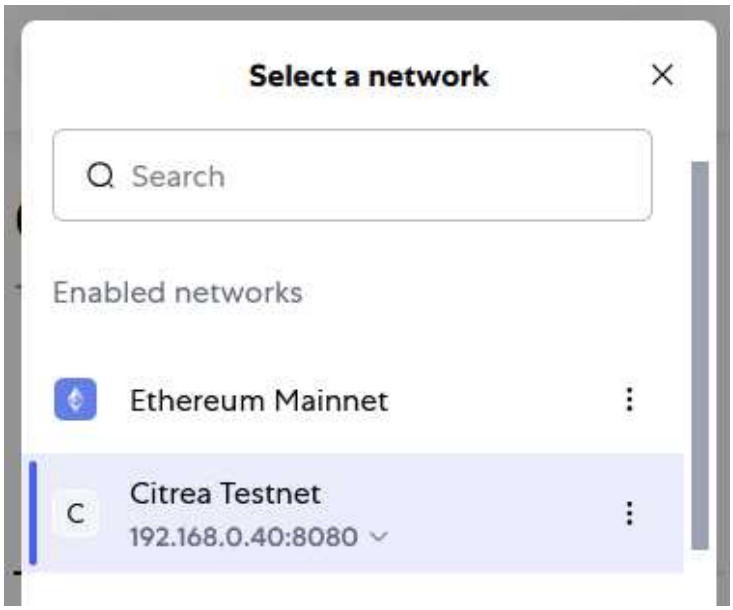
Citreaトークンは無い(売買しているCitreaトークンという意味だろう)

ブラウザにEVM walletをインストール

Windows Firefox で MetaMask をインストール

アカウント `0x181Bd70A1Bc17319630a7a182f48b869b21eb309`

ネットワークの設定は、ローカルでフルノードを立てていればそれを指定することができる。

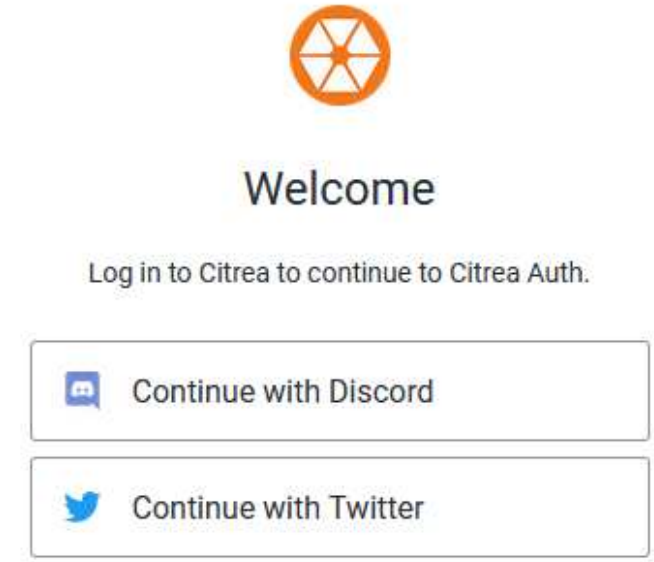


自分で立てていなくても、通常は faucetのサイトなど をブラウザで開いたまま MetaMask を起動すると自動で探してくれる。 `rpc.testnet.citrea.xyz` のようだ。



Citrea Faucetに接続

初回は認証がいるようだ。今回はDiscordにログインした。



explorer



MetaMask にアカウントをもう1つ作って送金

transaction

2秒くらいでconfirmしたということかな？

ローカルのフルノードが起動していないときに行うと結構待たされたので、たぶん同期が完了する必要がある。

エクスプローラを見るとUSDCとかあるが、お遊びで動かしているのだろうか？

testnet4 でもブリッジするしくみはあるのだろうが、やり方は分からなかった。

EVM walletでfaucetを受け取ることはできた(以下の「testnet4で動かす」参照)

10 BTCしか受け取れないということなので testnet4 で vout の value が 10 BTC のトランザクションを探した

tb1pkn7k07hjatwazlzkfku0ny6ksj6wa070tzjg8n4yq06tj799reesp6975aへの送金を行っているトランザクションはそうだと思う

N+1-of-N+1に10BTC送るtx

N+1-of-N+1送金をINPUTにしてN-of-Nに送金したtx



script

```
<witness>
OP_PUSHBYTES_32 <pubkey>
OP_CHECKSIG
OP_0
OP_IF
OP_PUSHBYTES_6 636974726561
OP_PUSHBYTES_20 564cb100d2d5deceb792fe913b9185fcfb808712
OP_PUSHBYTES_8 000000003b9aca00
OP_ENDIF
```

署名のチェックしかしてないな。

N-of-N送金をINPUTにしたtx

INPUTは2つあり、そのアドレスからのINPUTは普通のP2TRを解いただけのように見える。

もう1つの INPUT は `OP_CSV` があったのでちょっとそれっぽい

script

```
<witness>
OP_PUSHBYTES_2 c800
OP_CSV
```

OP_DROP

OP_PUSHBYTES_32

7c7b5c510f54bb7497609d90c88d67c296a01a0d353f7991a3de71d2e8d0ef1e

OP_CHECKSIG

最近だと2025/05/11のトランザクションもあった。

2025/04/21のブログには既にTangerineというネットワークにアップグレードしたように書かれていた。

ClementineがBitVMを完全サポートしたと書いてあるが、

regtestで動かす

testnet4 で Citrea の peg-in らしきトランザクションを見つけたが、Clementineが動作する前のものしかないからか見てもよくわからなかった(Citrea - Bitcoin L2#4a8687)。

2025/06/04 やってない

Taproot Recovery Address

Bitcoinチェーンからデポジットする話をしている？

開発

https://docs.citrea.xyz/developer-documentation/developer-documentation

RPC

けっこう少ない。

SoftConfirmation とか Slot とかよくわからん単語が出てきた