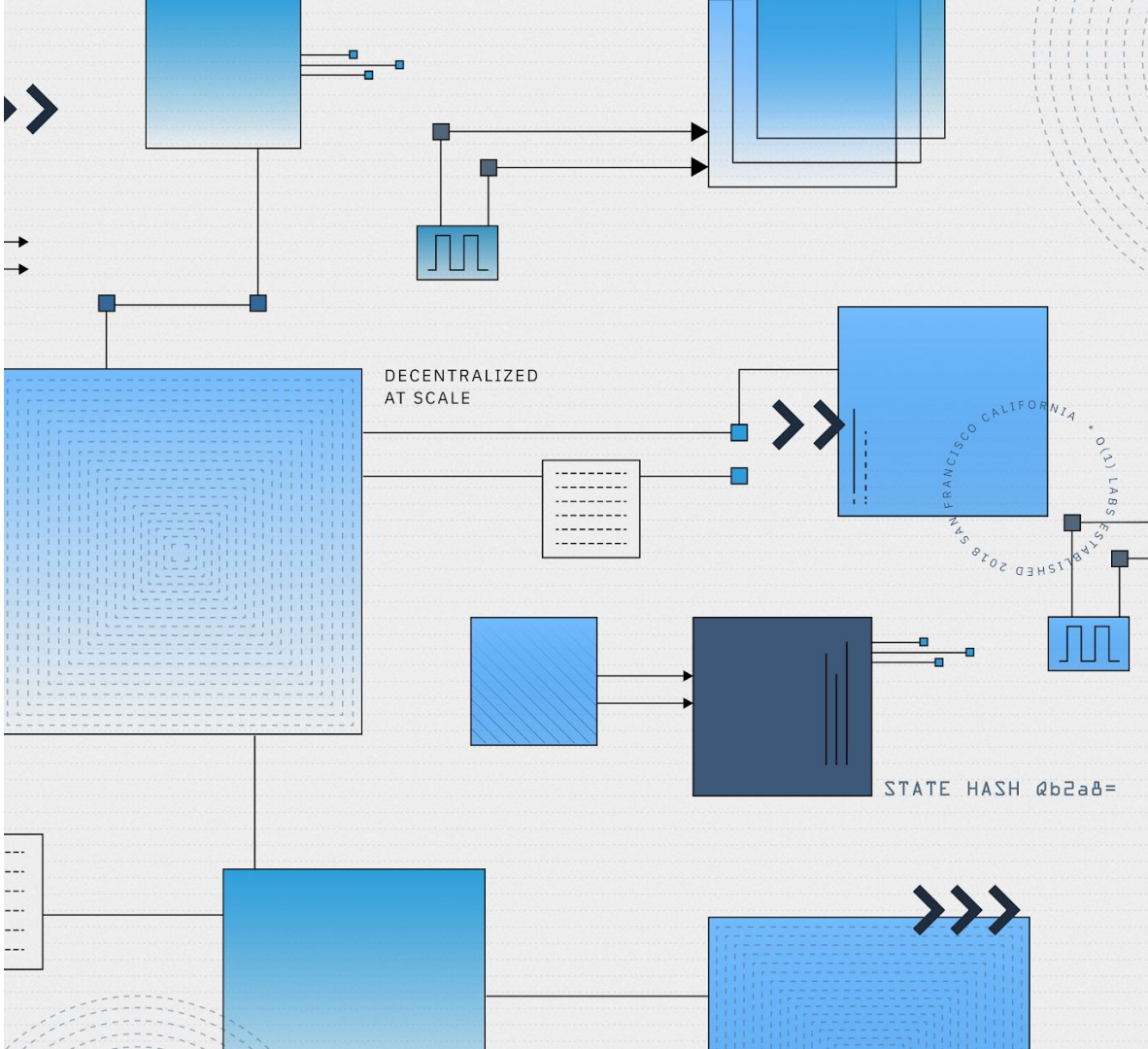
CODA

Decentralized at Scale

Avery



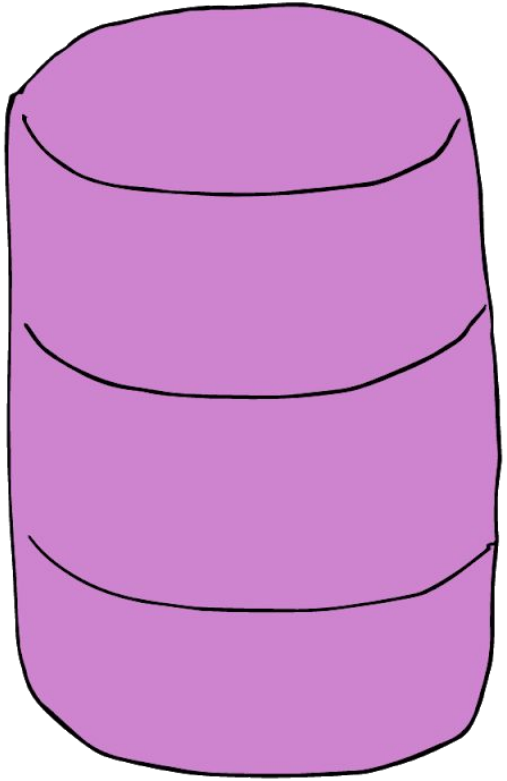
Coda's Tiny API

Blockchains get larger over time

Blockchains have gotten really huge

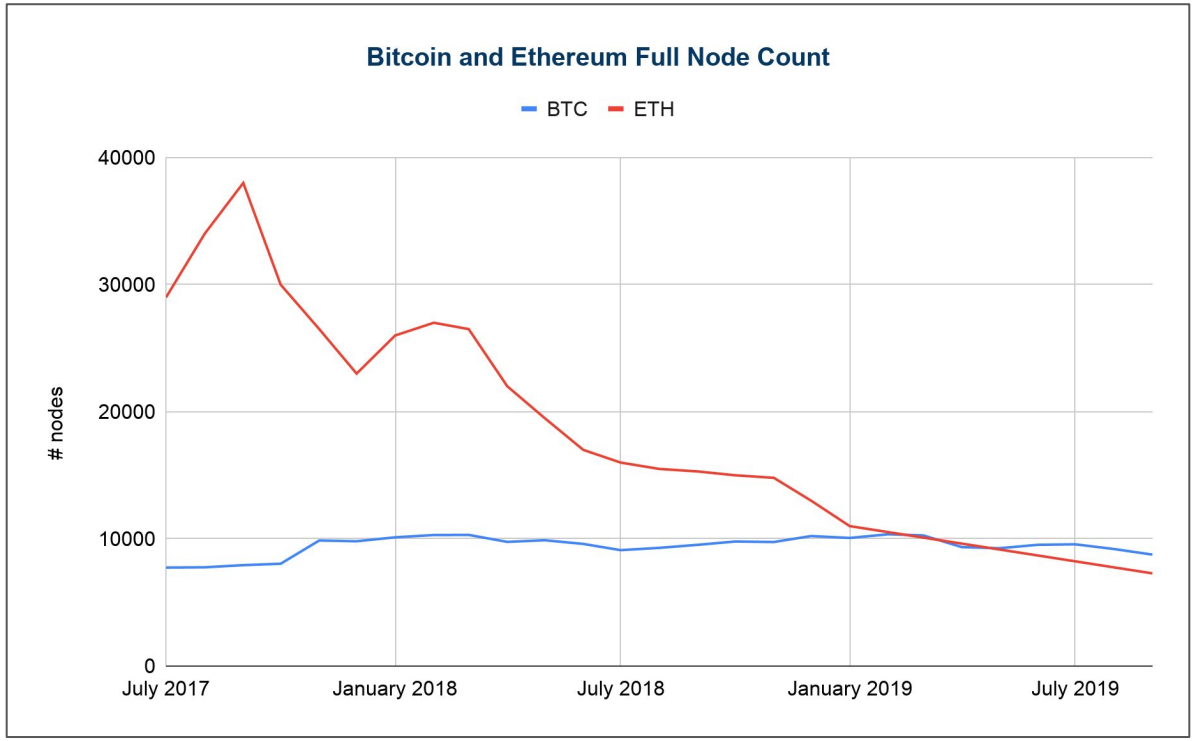
Individuals can’t download blockchains and run full nodes easily anymore.

This centralizes control in the hands of a shrinking set of entities that can afford to run nodes. This is bad for individuals to access cryptocurrency.



CODA

Blockchain size presents a challenge to decentralization



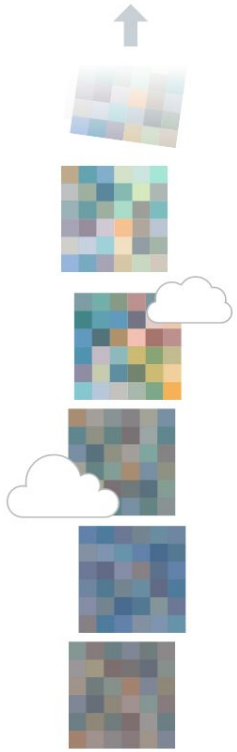
CODA

Coda

New Layer 1 Protocol

Replaces the blockchain with a recursively updating zkSNARK

- Everyone gets maximal full node security, even on phones and browsers
- SNARK is constant <1kB, independent of throughput
- Easy development: Library works everywhere, end-user's devices connect directly to the network



Coda
22kB

FIXED

Other blockchains
2TB+

INCREASING

CODA

zkSNARKs

Succinct
Non-interactive
ARgument
of
Knowledge

Small (<1 kB) proofs

We've been building tools to make writing snarks more like writing regular code, making possible large programs like Coda

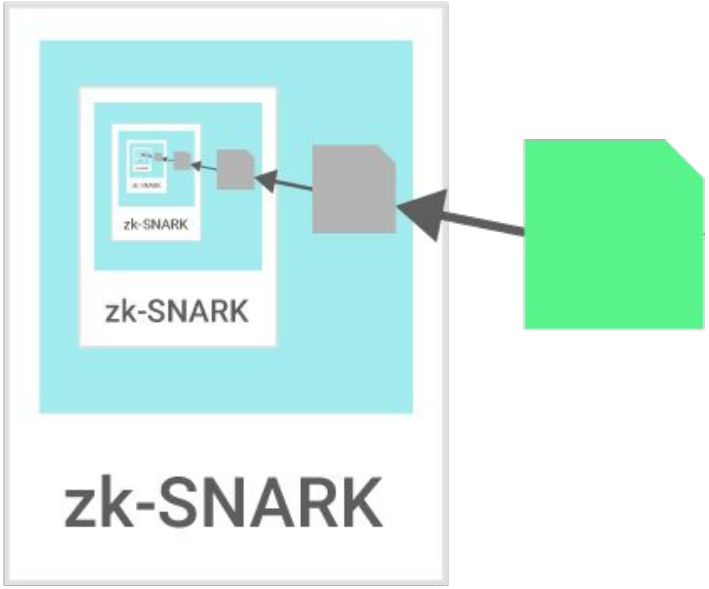
Our library is called **snarky**

Recursive Zero Knowledge Proofs

Instead of a blockchain: the network works together to recursively update a zkSNARK proof of the blockchain.

~1kB: Fast to share, check, and is an equivalent computation to downloading and checking the whole blockchain.

Recursion means efficient to update. The new proof can recursively build off of its predecessor.



CODA

Our Vision for Coda's Developer Experience

- With Coda:
  - Import a full node directly into your program, including websites, no 3rd parties or browser extensions. The Coda library will connect directly to the network from code on users' devices. SNARKs for privacy.

- Today's Testnet
  - Working succinct blockchain
  - GraphQL API
  - macOS (brew) and Linux (apt-get)

```
<script src="coda_api.js"></script>
<script>
  onClick(button)
    .then(() => Coda.requestWallet())
    .then((wallet) =>
        wallet.sendTransaction(...))
</script>
```

CODA

Hackathon Prize

Painting with Coda ($2,500)

- Must use the Coda GraphQL API
- Visualize some aspect of the network

Connect to the Testnet

- Connect to the latest testnet
- Request funds from the faucet and send a transaction!
- Show us at our booth for LIMITED EDITION swag

https://bit.ly/CodaHackathon

CODA

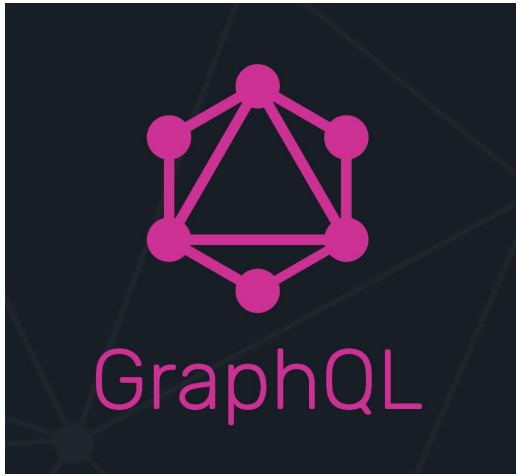
GraphQL Workshop

CODA

What is GraphQL?

Improved API query language

- Typed
- Discoverable
- Lots of existing tooling
- Plays well with React
- Support for subscriptions



http://graphql.o1test.net/graphql

Public read-only API

CODA

Playing with GraphiQL

- Visual API explorer
- Easily prototype queries
- See data response
- Inline docs
- Generate code!

Explorer

query MyQuery
- account
- blockchainVerificationKey
- blocks
  - after:
  - before:
  - filter:
  - first: 2
  - last:
  - edges
  - nodes
    - creator
    - creatorAccount
    - protocolState
      - blockchainState
      - consensusState
      - previousStateHash
    - protocolStateProof
    - snarkJobs
    - stateHash
    - stateHashField
    - transactions
  - pageInfo
  - totalCount
- daemonStatus
- pooledUserCommands
- snarkPool
- syncStatus
- trackedAccounts
- transactionStatus
- trustStatus
- trustStatusAll
- version

+ ADD NEW QUERY

+ ADD NEW SUBSCRIPTION

GraphiQL | Prettify | History | Explorer | Exporter | Docs

```
query MyQuery {
  version
  blocks(first: 2) {
    nodes {
      creator
      protocolState {
        previousStateHash
      }
    }
  }
}
```

QUERY VARIABLES


```
{
  "data": {
    "version": "6b47cff2cd168803d72cbb86e424def4655ea0ba",
    "blocks": {
      "nodes": [
        {
          "creator":
 "tdNDjjAwyjgvNgU1x8uWs1n44H8vK7UGrRLvXzkcv8iC3Jw9A1B1UDUZndg
BZh6zTb23pNatt7ujfTbCVjiihTZRMJcErZSkz93qE5Ue5VpAJsvaQvpHQGj3
XexP2fK6i6xMfQArSvcXWV",
          "protocolState": {
            "previousStateHash":
 "4ApWN4yFVvw1RakRhCfj7ZvhDbQXXL9zhAMiWbnhVZ4CZphqYhwu44Ef2yr
GFxX3orEhmrzfPhKGv6Zc3hNRgvSGwfVC8nxgZ5JcysHBbURvG4jkBwe4CtQM
2sqzCr5W1CQRQr8w3hruH"
          }
        },
        {
          "creator":
 "tdNDs39Vk2rLLy9o43yNVBFrouEY9p49SHQjSyV4vvX7wpi52x1k3Eykg8s
oVmwLFDqJBfF3vvKWZuz2xjaQT9opUA5P2miTpJwP1RGu9vm2mM1gFcZxNd1f
HBY8Jggezr7We9qCShZwCY",
          "protocolState": {
            "previousStateHash":
 "4ApWJfCTvCmMMippR4e6Z6VMMtwmhDziPvpAx8UhPrUeLNBGpiL5CEL15Jn
gJopuKMCaSpJN941z7T2GXS1ySSbxqD4Da7hFpqjYPoZh3ykZ3sxdg7zWBpdB
beJXZ9rigbyKhsAdfZRdW"
          }
        },
        {
          "creator":
 "tdNDjjAwyjgvNgU1x8uWs1n44H8vK7UGrRLvXzkcv8iC3Jw9A1B1UDUZndg
BZh6zTb23pNatt7ujfTbCVjiihTZRMJcErZSkz93qE5Ue5VpAJsvaQvpHQGj3
XexP2fK6i6xMfQArSvcXWV",
          "protocolState": {
            "previousStateHash":
 "4ApWK7UzBsefJSdPkCQ77cDP6sqtg6VdwF6hknLNYe7MhqnTJfDjZgnudHc
Ct8F4SsG2RTWzPFUpHP836dY6VySPK8fxQBJZ4moPhBGzpXT4Rm1bF52fEDTd
h3msVW93whEjAhqiGCrNB"
          }
        },
        {
          "creator":
 "tdNDypcUkVoNSsbdhT9ytXwVxxU2ECEhaLsgq2ceiBzRxQon8txsofVs9BA
jb43BrWf5rUs5ewRyogkiaWddSkGNhXQHCk8FtJ4HDUfPARHMYstvEP2h9CWu
E5t1MdUD1Q9tXDPJRMy2qq",
          "protocolState": {
            "previousStateHash":
 "4ApWMzC33UxJ7vdGKFdeKRbGRENw47iYimhKTJM7z3tDQRKzqxbPXxVymja
aHw8i7piX4YFr2G1xqjFNmPepK596vVrZjymBh8rJKZwTk66kec6K91Nfm4WU
1HMZbE3vudGWdGT7zYa7N"
```


CODA

Your first query

```
query MyFirstQuery {
  version
  daemonStatus {
    blockchainLength
    numAccounts
  }
  syncStatus
}
```


```
{
  "data": {
    "version": "6b47cff2cd168803d72cbb86e424def4655ea0ba",
    "daemonStatus": {
      "blockchainLength": 161,
      "numAccounts": 109
    },
    "syncStatus": "SYNCED"
  }
}
```


CODA

Step 1: Setting up your React App

```
npx create-react-app coda-viz
cd coda-viz
code . (or your prefered editor)
npm start
```

CODA

Step 2: Use Code Exporter

Docs

Code Exporter

JavaScript

fetch

fetch

react-apollo

OPTIONS

server-side usa

async/await

55ea0ba",

```
/*
This is an example snippet - you should consider tailoring it
to your service.
*/

async function fetchGraphQL(operationsDoc, operationName, varia
  const result = await fetch(
    "http://graphql.o1test.net:10900/graphql",
    {
```

CODA

Step 3: Copy-paste into `src/index.js`

index.js — coda-viz

EXPLORER

OPEN EDITORS

index.js src M

CODA-VIZ

node_modules

public

src

App.css

App.js

App.test.js

index.css

index.js M

logo.svg

serviceWorker.js

.gitignore

package-lock.json U

package.json M

README.md

yarn.lock

src > index.js > ...

```
/* This is an example snippet - you should consider tailoring it
to your service.
*/
/*
  Add these to your `package.json`:
    "apollo-boost": "^0.3.1",
    "graphql": "^14.2.1",
    "graphql-tag": "^2.10.0",
    "react-apollo": "^2.5.5"
*/

import gql from "graphql-tag";
import React from "react";
import ReactDOM from "react-dom";
import { ApolloClient, InMemoryCache, HttpLink } from "apollo-boost";
import { Query, ApolloProvider } from "react-apollo";

// This setup is only needed once per application;
const apolloClient = new ApolloClient({
  cache: new InMemoryCache(),
  link: new HttpLink({
    uri: "http://graphql.o1test.net:10900/graphql",
  }),
});

const MY_FIRST_QUERY_QUERY = gql`
  query MyFirstQuery {
```

CODA

Step 4: Update your dependencies

Copy dependencies into package.json

```
npm install
```

Step 5: See the result!

```
npm start
```

localhost:3000


```
{
  "version": "6b47cff2cd168803d72cbb86e424def4655ea0ba",
  "daemonStatus": {
    "blockchainLength": 165,
    "numAccounts": 109,
    "__typename": "DaemonStatus"
  },
  "syncStatus": "SYNCED"
}
```


CODA

Start hacking!

CODA

https://bit.ly/CodaHackathon

CODA

Technical Community : https://bit.ly/CodaDiscord
Follow our progress : @CodaProtocol
Website : https://codaprotocol.com