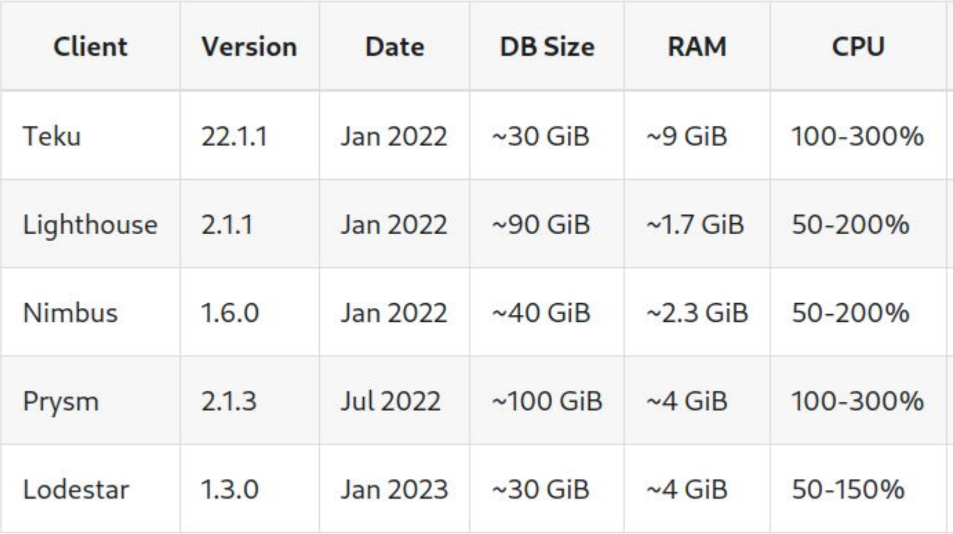
Portal CTI Network: Trin Implementation

Daniel Ramirez Chiquillo
Ethereum Protocol Fellowship Cohort 4

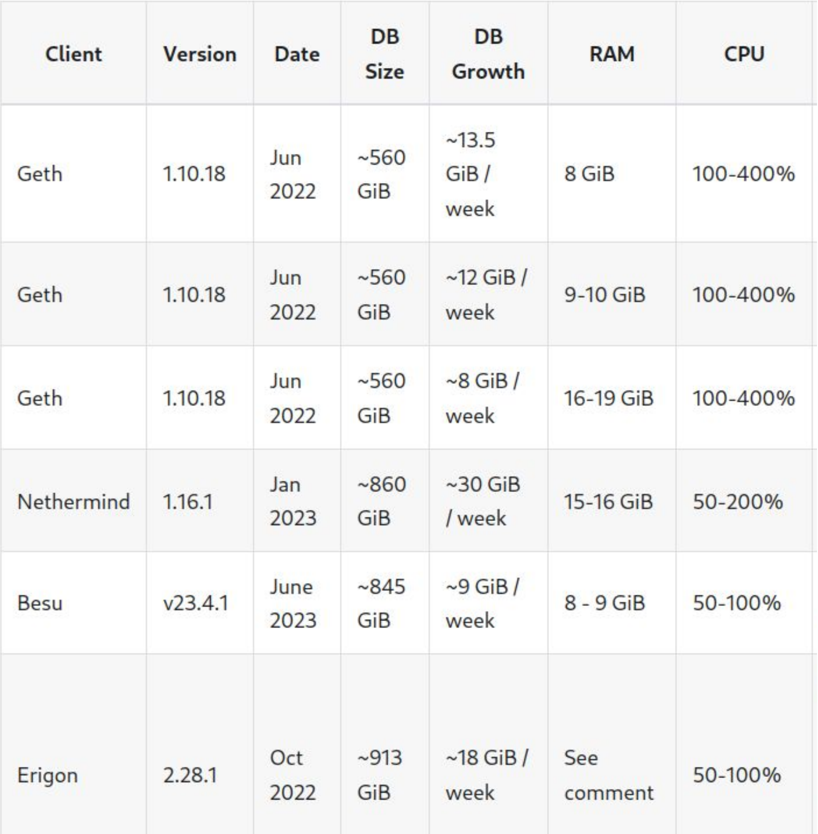
2023-08-08

Current Options

Running a Full Node



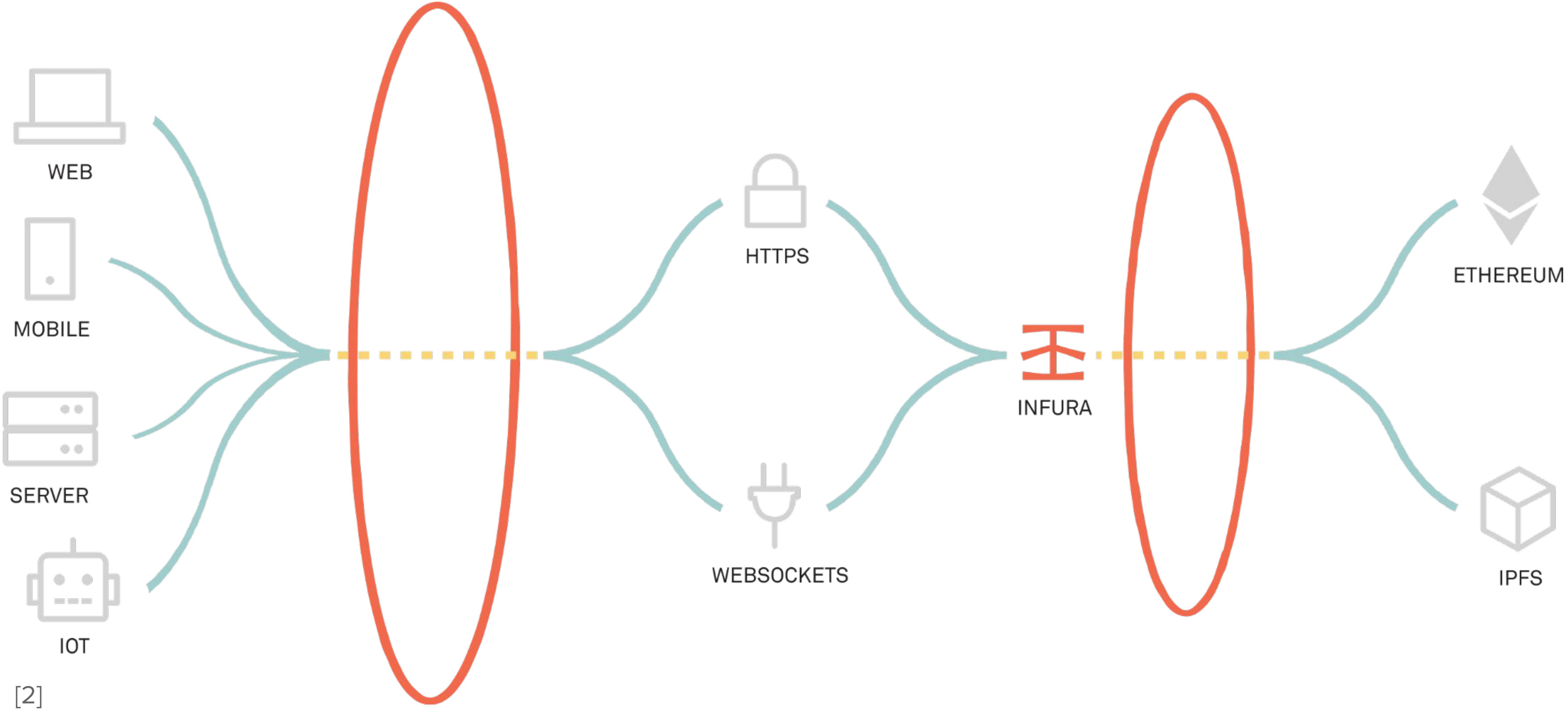
Client	Version	Date	DB Size	RAM	CPU
Teku	22.1.1	Jan 2022	~30 GiB	~9 GiB	100-300%
Lighthouse	2.1.1	Jan 2022	~90 GiB	~1.7 GiB	50-200%
Nimbus	1.6.0	Jan 2022	~40 GiB	~2.3 GiB	50-200%
Prysm	2.1.3	Jul 2022	~100 GiB	~4 GiB	100-300%
Lodestar	1.3.0	Jan 2023	~30 GiB	~4 GiB	50-150%



Client	Version	Date	DB Size	DB Growth	RAM	CPU
Geth	1.10.18	Jun 2022	~560 GiB	~13.5 GiB / week	8 GiB	100-400%
Geth	1.10.18	Jun 2022	~560 GiB	~12 GiB / week	9-10 GiB	100-400%
Geth	1.10.18	Jun 2022	~560 GiB	~8 GiB / week	16-19 GiB	100-400%
Nethermind	1.16.1	Jan 2023	~860 GiB	~30 GiB / week	15-16 GiB	50-200%
Besu	v23.4.1	June 2023	~845 GiB	~9 GiB / week	8 - 9 GiB	50-100%
Erigon	2.28.1	Oct 2022	~913 GiB	~18 GiB / week	See comment	50-100%

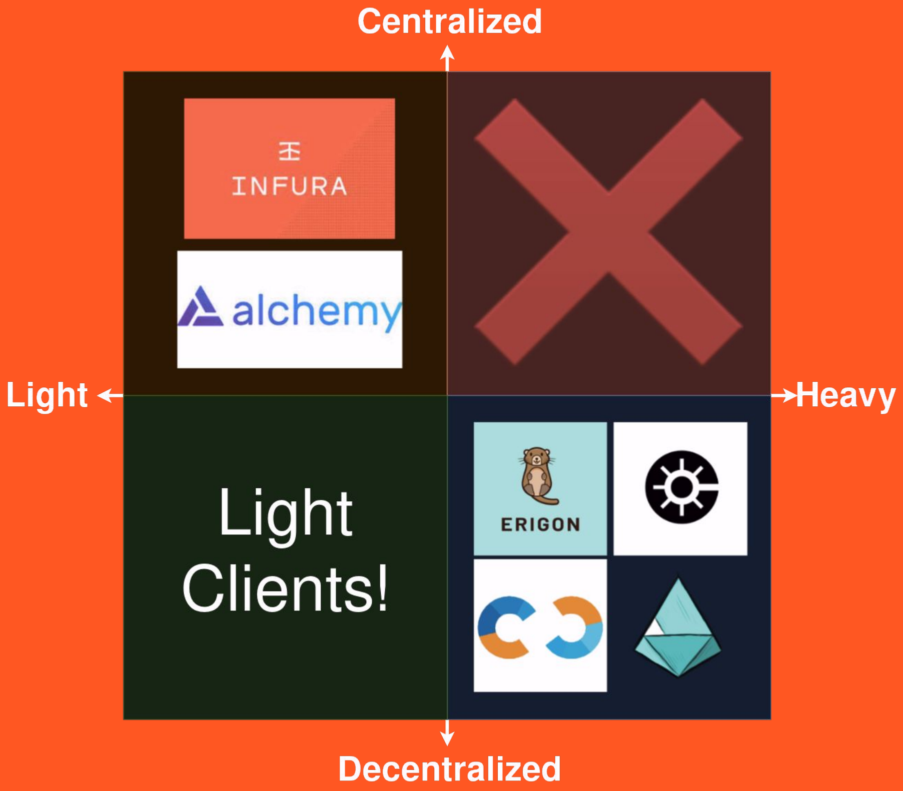
[1]

Relying on a Centralized Provider



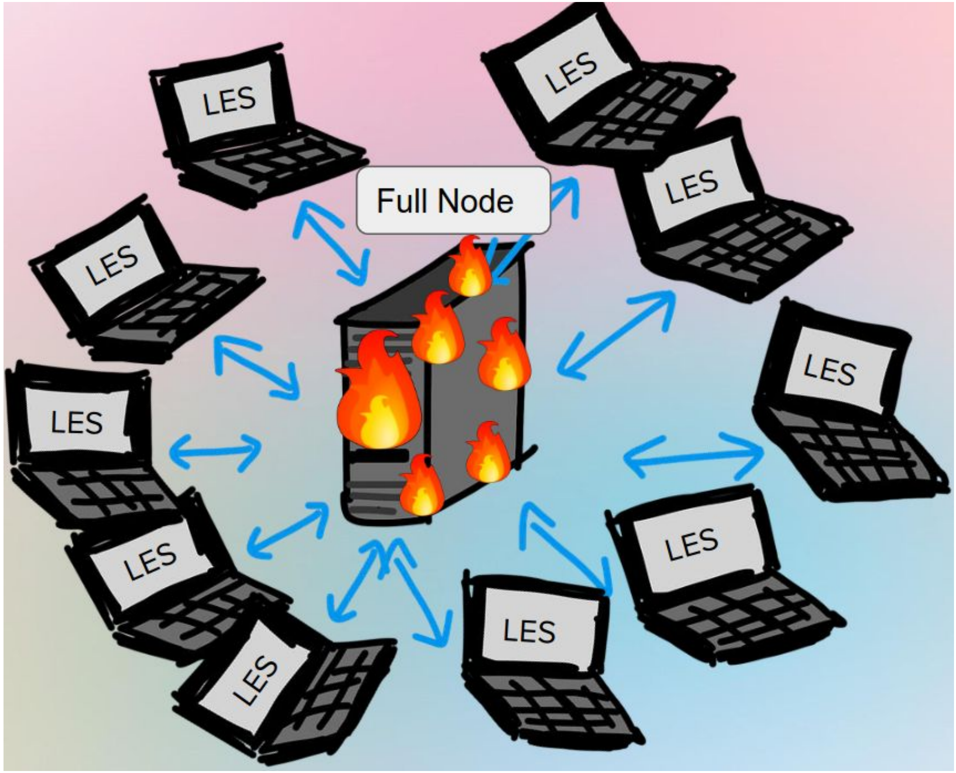
[2]

We want lightweight decentralized access to the Ethereum network



Light Clients

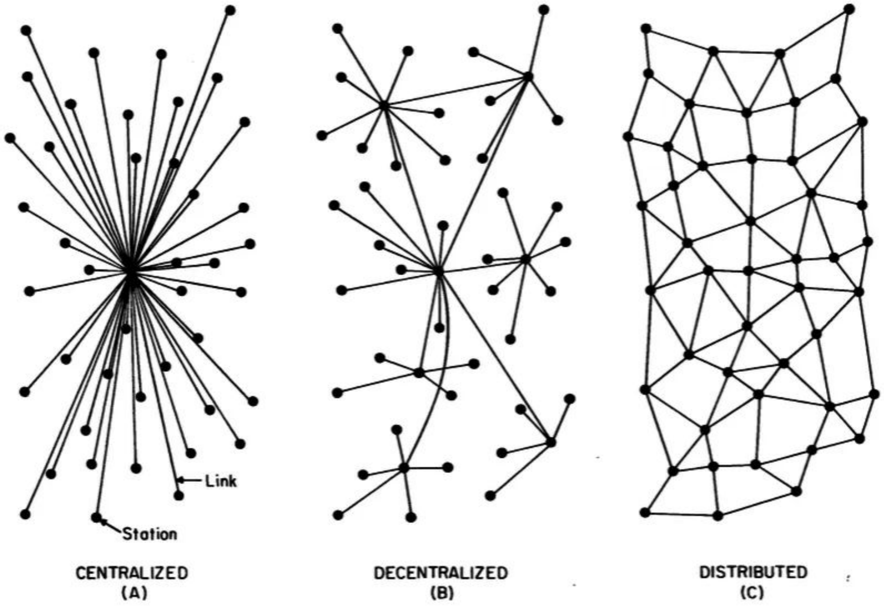
Light Ethereum Subprotocol (LES)



[3]

- Built on devp2p.
- Server-client architecture.
- Decentralized but not distributed.
- Doesn’t scale.

The Portal Network



- Built on Discovery V5.
- All nodes are equal.
- Distributed.
- Gets better with more nodes.

[4]

The Portal Clients

Trin	Rust
Ultralight	Typescript
Fluffy	Nim

[4]

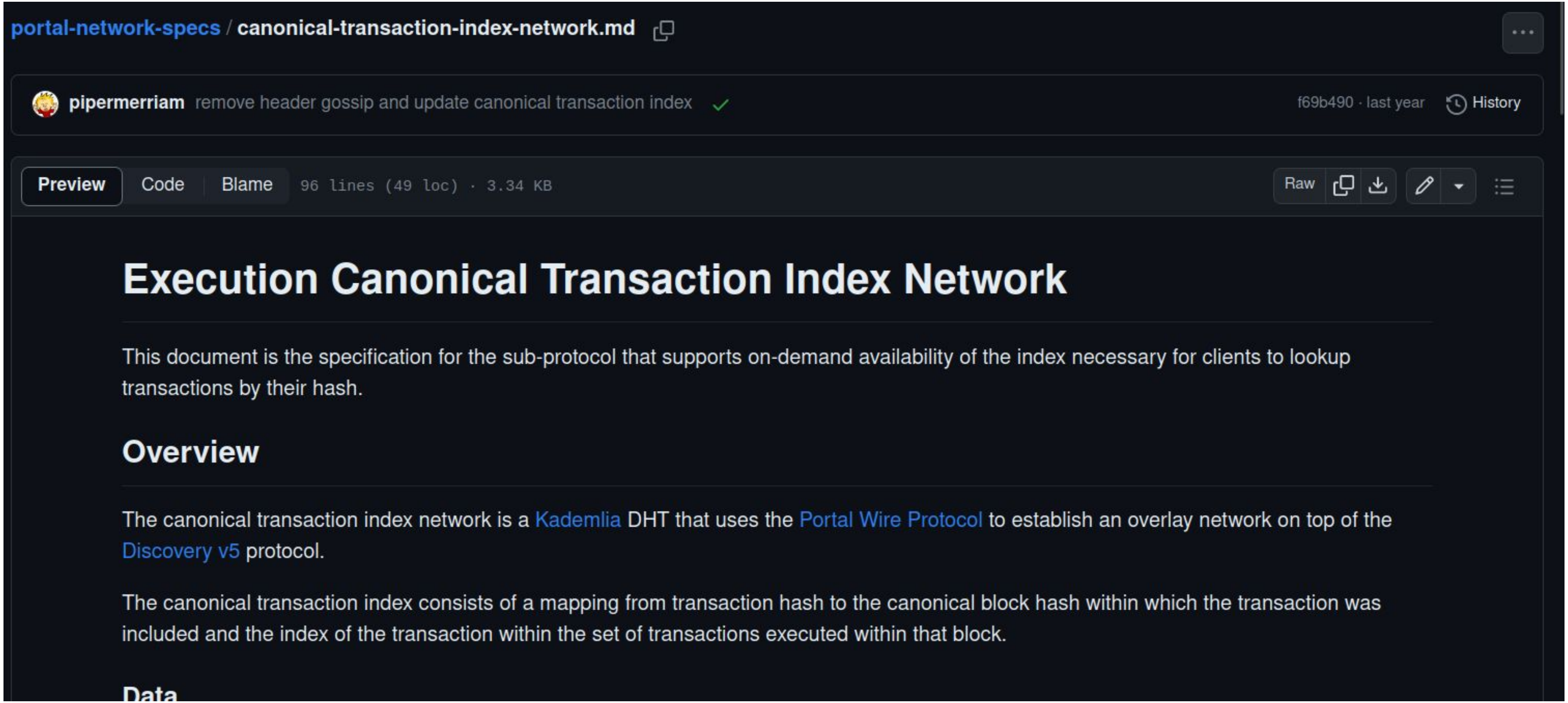
The Portal Sub-Protocols

Network	Data	Status (Trin)
History Network	Headers, block bodies, receipts	Almost ready
Beacon light client	Beacon chain light protocol data	Implementing
State network	Account and contract storage	Re-started
Canonical txn index	Transaction data	Spec ready
Transaction Gossip	Lightweight mempool	WIP spec

[4]

Implementing the CTI Network

Specs



portal-network-specs / canonical-transaction-index-network.md
pipermerriam remove header gossip and update canonical transaction index ✓
f69b490 · last year
History
Preview
Code
Blame
96 lines (49 loc) · 3.34 KB
Raw

Execution Canonical Transaction Index Network

This document is the specification for the sub-protocol that supports on-demand availability of the index necessary for clients to lookup transactions by their hash.

Overview

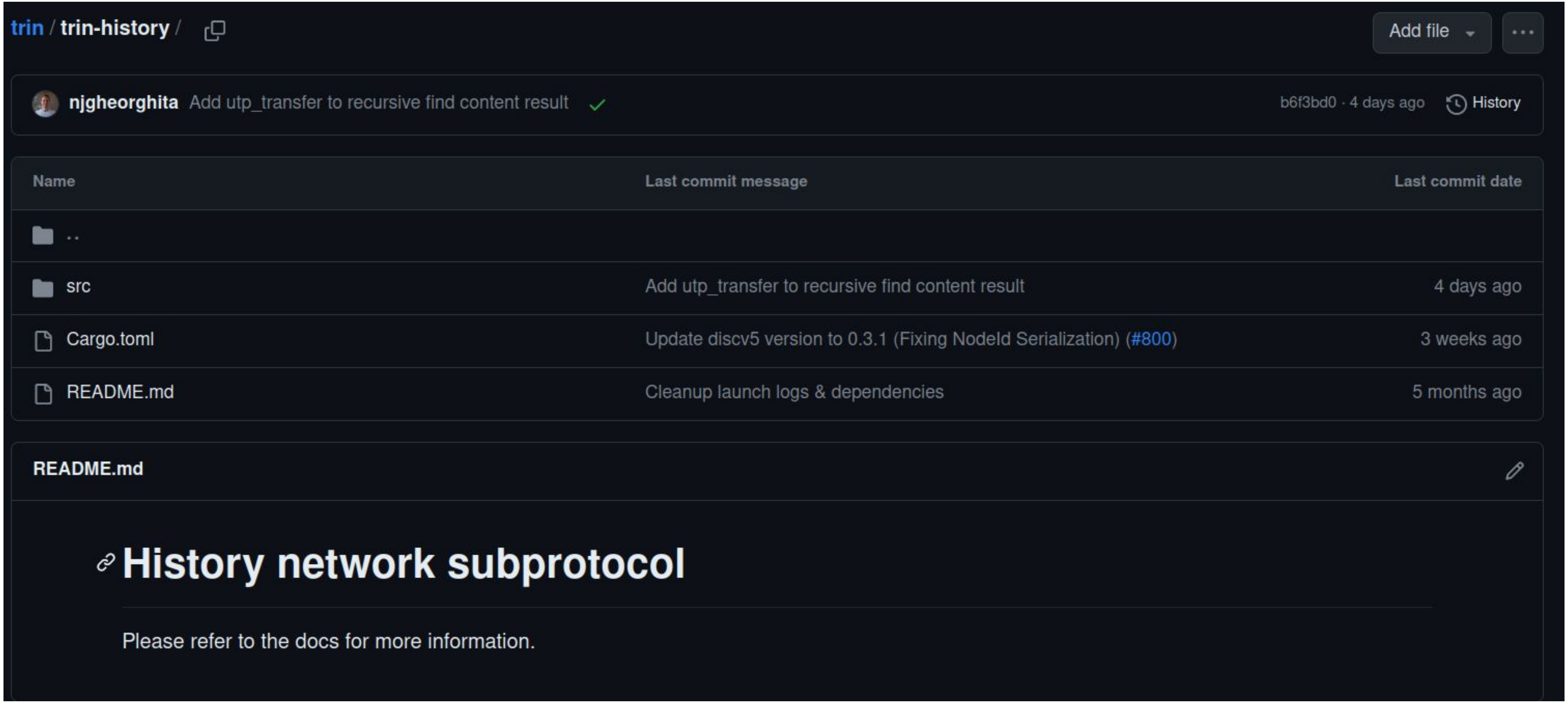
The canonical transaction index network is a Kademlia DHT that uses the Portal Wire Protocol to establish an overlay network on top of the Discovery v5 protocol.

The canonical transaction index consists of a mapping from transaction hash to the canonical block hash within which the transaction was included and the index of the transaction within the set of transactions executed within that block.

Data

[5]

Other Networks



trin / trin-history /

njgheorghita Add utp_transfer to recursive find content result ✓ b6f3bd0 · 4 days ago

Name	Last commit message	Last commit date
..		
src	Add utp_transfer to recursive find content result	4 days ago
Cargo.toml	Update discv5 version to 0.3.1 (Fixing NodeId Serialization) (#800)	3 weeks ago
README.md	Cleanup launch logs & dependencies	5 months ago

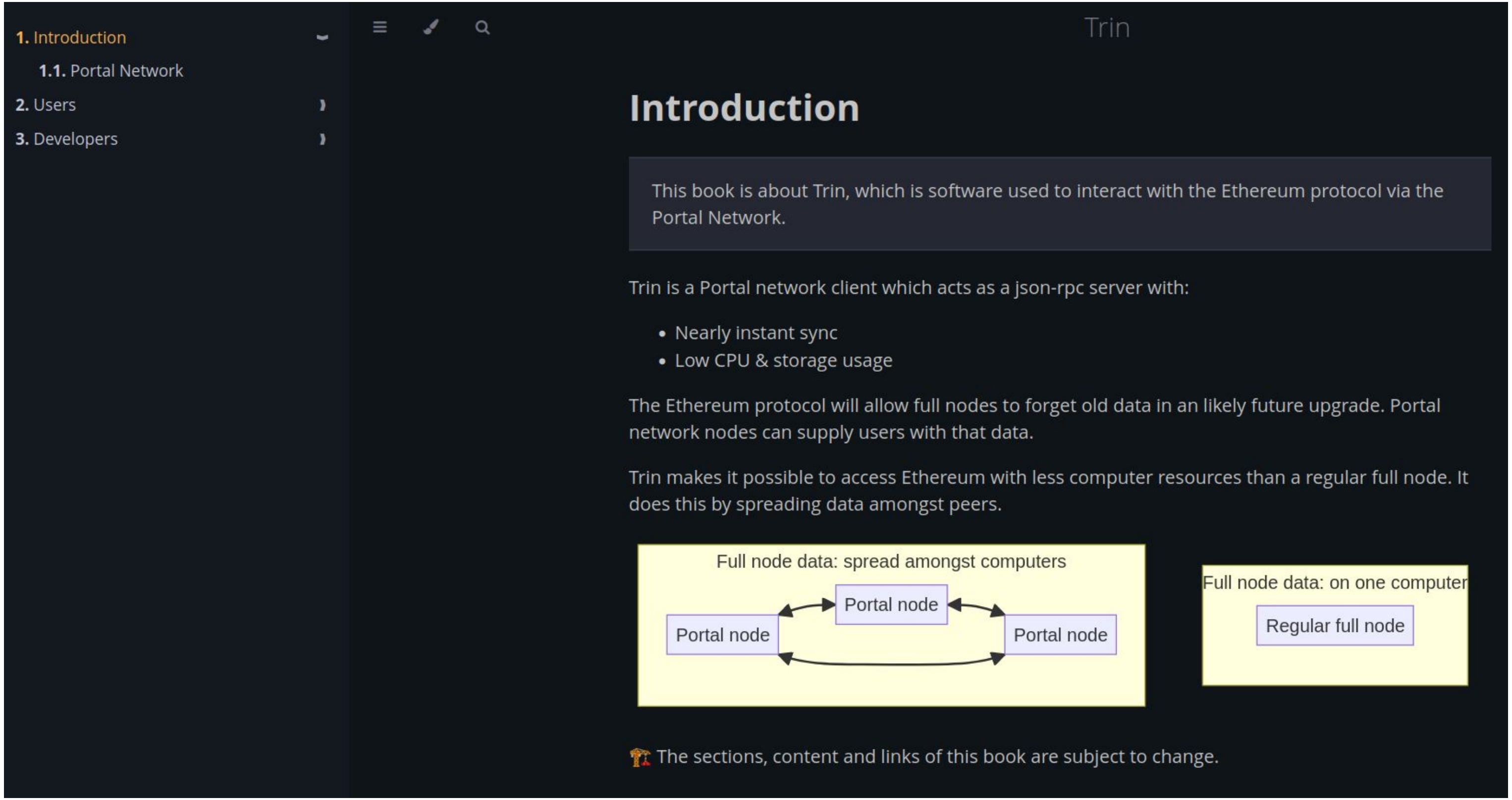
README.md

History network subprotocol

Please refer to the docs for more information.

[6]

Book



Introduction

> This book is about Trin, which is software used to interact with the Ethereum protocol via the Portal Network.

Trin is a Portal network client which acts as a json-rpc server with:

- Nearly instant sync
- Low CPU & storage usage

The Ethereum protocol will allow full nodes to forget old data in an likely future upgrade. Portal network nodes can supply users with that data.

Trin makes it possible to access Ethereum with less computer resources than a regular full node. It does this by spreading data amongst peers.

Full node data: spread amongst computers
Portal node
Portal node
Portal node
Full node data: on one computer
Regular full node

🏗️ The sections, content and links of this book are subject to change.

Roadmap

Weeks 1-2	Week 3	Weeks 4-10	Week 11	Week 12
Learn	Design	Code	Test	Document

Challenges

- Possible changes in the CTIN spec.
- Estimating the time it would take to code the network.
- Possible changes in the Portal Network design.

Goals

- Main goal: Contribute to the portal network.
- Implement the Canonical Transaction Index Network.
- Improve documentation.
- Learn more about the portal network.

Sources

[1] https://eth-docker.net/Usage/ResourceUsage/

[2] https://consensys.net/blog/news/why-infura-is-the-secret-weapon-of-ethereum-infrastructure/

[3] https://archive.devcon.org/resources/6/the-portal-network.pdf

[4] https://berty.tech/blog/decentralized-distributed-centralized

[5] https://github.com/ethereum/portal-network-specs/blob/master/canonical-transaction-index-network.md

[6] https://github.com/ethereum/trin/tree/master/trin-history

[7] https://ethereum.github.io/trin/

Thanks!