Less State, More Logs!

~~~~~~~~~~~~~~~~~~~~

Scaling Ethereum With A Log-Based Storage Architecture

Zsolt Felföldi

Ethereum Foundation (GETH)

zsfelfoldi@ethereum.org
~~~~~~~~~~~~~~~~~~~~

Trustless Execution Layer API

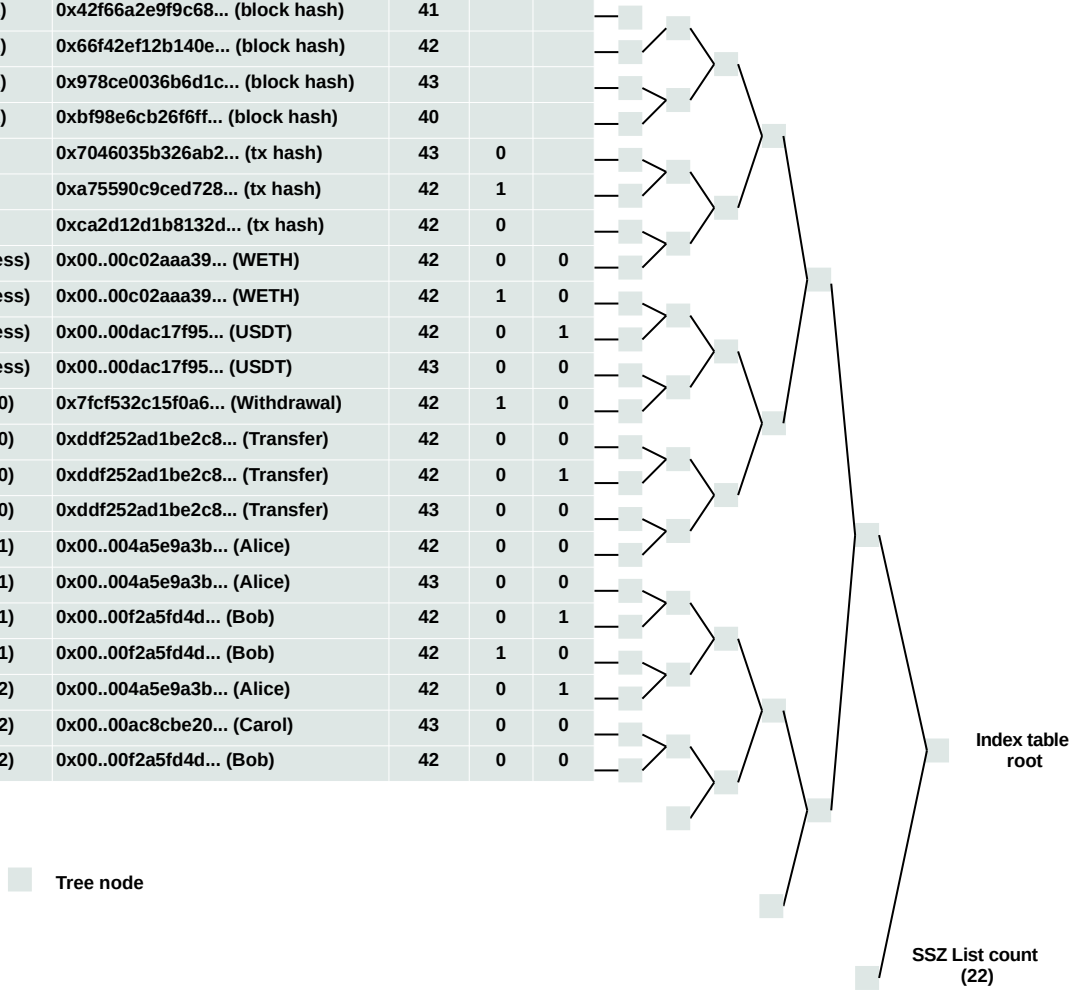
JSON RPC	Trustless execution layer REST API
`eth_blockNumber`	none (assumed to be known by the client)
`eth_getBalance`	`/eth/v1/exec/state`
`eth_getBlockByHash`	`/eth/v1/exec/blocks` if canonicalness needs to be proven, `/eth/v1/exec/history` * *alternative: `/eth/v1/exec/headers` from queried block to head
`eth_getBlockByNumber`	`/eth/v1/exec/blocks` , `/eth/v1/exec/history` * *alternative: `/eth/v1/exec/headers` from queried block to head
`eth_getBlockReceipts`	`/eth/v1/exec/block_receipts`
`eth_getBlockTransactionCountByHash`	`/eth/v1/exec/transaction_by_index`
`eth_getBlockTransactionCountByNumber`	`/eth/v1/exec/history` * or `/eth/v1/exec/headers` from queried block to head `/eth/v1/exec/transaction_by_index`
`eth_getCode`	`/eth/v1/exec/state`
`eth_getLogs`	`/eth/v1/exec/logs` * *alternative: `/eth/v1/exec/headers` from range start to head, `/eth/v1/exec/block_receipts` for all bloom filter matches
`eth_getProof`	`/eth/v1/exec/state`
`eth_getStorageAt`	`/eth/v1/exec/state`
`eth_getTransactionByBlockHashAndIndex`	`/eth/v1/exec/headers` , `/eth/v1/exec/transaction_by_index`
`eth_getTransactionByBlockNumberAndIndex`	`/eth/v1/exec/history` *, `/eth/v1/exec/headers` for queried block, `/eth/v1/exec/transaction_by_index` *alternative: `/eth/v1/exec/headers` from queried block to head
`eth_getTransactionByHash`	`/eth/v1/exec/transaction` `/eth/v1/exec/transaction_position` * if proof of canonical inclusion position is needed *alternative for canonical transactions: `/eth/v1/exec/headers` from inclusion block to head, `/eth/v1/exec/transaction_by_index`

- Endpoint format similar to the beacon chain REST API
- Works together with a beacon light client; answers all queries with Merkle proofs based on latest execution block hash
- New functionality based on the Trustless Log Index (EIP-7745B)

See full draft proposal at:

https://github.com/zsfelfoldi/tli

Type	Value	Block #	Tx #	Log #
0 (Block)	0x42f66a2e9f9c68... (block hash)	41		
0 (Block)	0x66f42ef12b140e... (block hash)	42		
0 (Block)	0x978ce0036b6d1c... (block hash)	43		
0 (Block)	0xbf98e6cb26f6ff... (block hash)	40		
1 (Tx)	0x7046035b326ab2... (tx hash)	43	0	
1 (Tx)	0xa75590c9ced728... (tx hash)	42	1	
1 (Tx)	0xca2d12d1b8132d... (tx hash)	42	0	
2 (Address)	0x00..00c02aaa39... (WETH)	42	0	0
2 (Address)	0x00..00c02aaa39... (WETH)	42	1	0
2 (Address)	0x00..00dac17f95... (USDT)	42	0	1
2 (Address)	0x00..00dac17f95... (USDT)	43	0	0
3 (Topic0)	0x7fcf532c15f0a6... (Withdrawal)	42	1	0
3 (Topic0)	0xddf252ad1be2c8... (Transfer)	42	0	0
3 (Topic0)	0xddf252ad1be2c8... (Transfer)	42	0	1
3 (Topic0)	0xddf252ad1be2c8... (Transfer)	43	0	0
4 (Topic1)	0x00..004a5e9a3b... (Alice)	42	0	0
4 (Topic1)	0x00..004a5e9a3b... (Alice)	43	0	0
4 (Topic1)	0x00..00f2a5fd4d... (Bob)	42	0	1
4 (Topic1)	0x00..00f2a5fd4d... (Bob)	42	1	0
5 (Topic2)	0x00..004a5e9a3b... (Alice)	42	0	1
5 (Topic2)	0x00..00ac8cbe20... (Carol)	43	0	0
5 (Topic2)	0x00..00f2a5fd4d... (Bob)	42	0	0



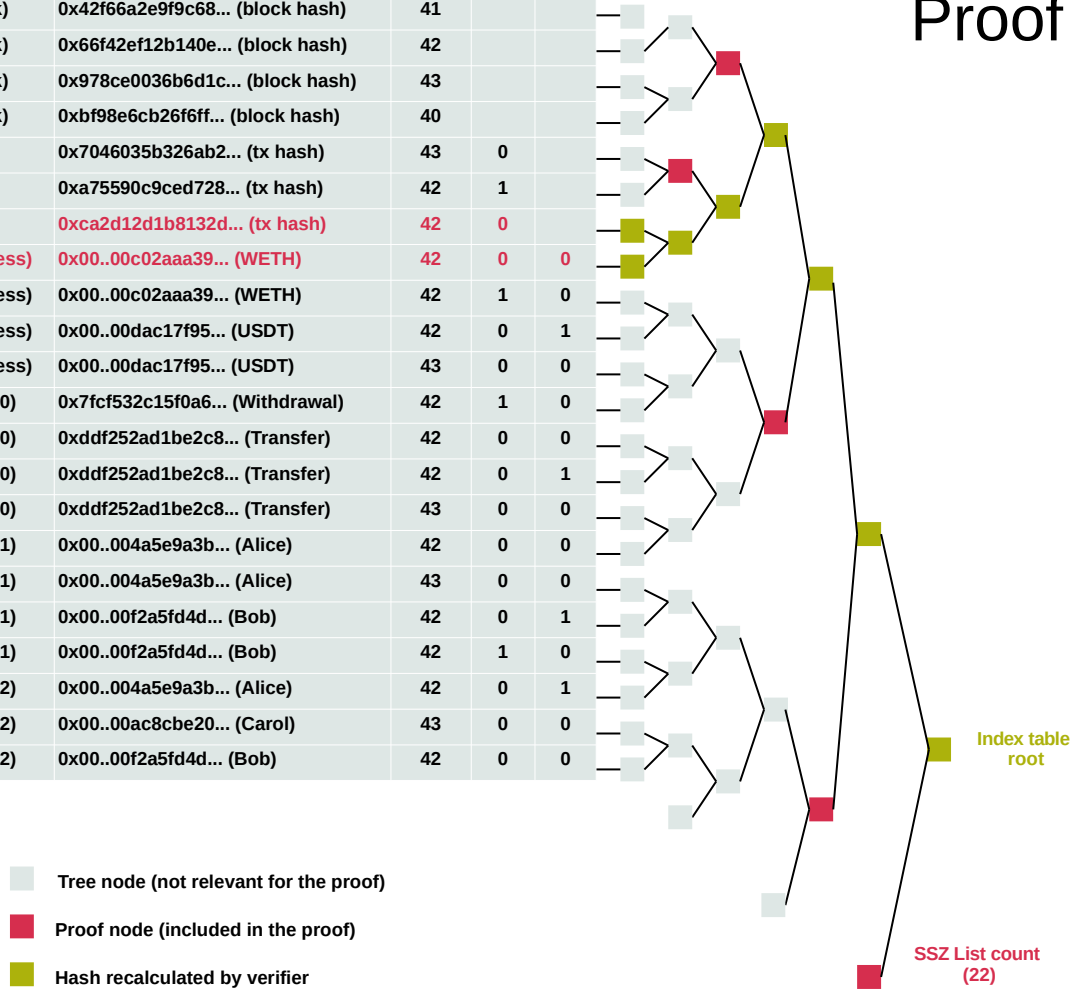
The Trustless Log Index

(introducing index tables)

- A lexicographically ordered list of entries in a given block range
  - Log addresses and topics
  - Block hashes, transaction hashes
- Stored with position information
- Hashed into a Binary Merkle Tree
- Inclusion and exclusion proofs for specific entries or search patterns

Proof example: exclusion proof

Type	Value	Block #	Tx #	Log #
0 (Block)	0x42f66a2e9f9c68... (block hash)	41		
0 (Block)	0x66f42ef12b140e... (block hash)	42		
0 (Block)	0x978ce0036b6d1c... (block hash)	43		
0 (Block)	0xbf98e6cb26f6ff... (block hash)	40		
1 (Tx)	0x7046035b326ab2... (tx hash)	43	0	
1 (Tx)	0xa75590c9ced728... (tx hash)	42	1	
1 (Tx)	0xca2d12d1b8132d... (tx hash)	42	0	
2 (Address)	0x00..00c02aaa39... (WETH)	42	0	0
2 (Address)	0x00..00c02aaa39... (WETH)	42	1	0
2 (Address)	0x00..00dac17f95... (USDT)	42	0	1
2 (Address)	0x00..00dac17f95... (USDT)	43	0	0
3 (Topic0)	0x7fcf532c15f0a6... (Withdrawal)	42	1	0
3 (Topic0)	0xddf252ad1be2c8... (Transfer)	42	0	0
3 (Topic0)	0xddf252ad1be2c8... (Transfer)	42	0	1
3 (Topic0)	0xddf252ad1be2c8... (Transfer)	43	0	0
4 (Topic1)	0x00..004a5e9a3b... (Alice)	42	0	0
4 (Topic1)	0x00..004a5e9a3b... (Alice)	43	0	0
4 (Topic1)	0x00..00f2a5fd4d... (Bob)	42	0	1
4 (Topic1)	0x00..00f2a5fd4d... (Bob)	42	1	0
5 (Topic2)	0x00..004a5e9a3b... (Alice)	42	0	1
5 (Topic2)	0x00..00ac8cbe20... (Carol)	43	0	0
5 (Topic2)	0x00..00f2a5fd4d... (Bob)	42	0	0



Block range: #40..#43

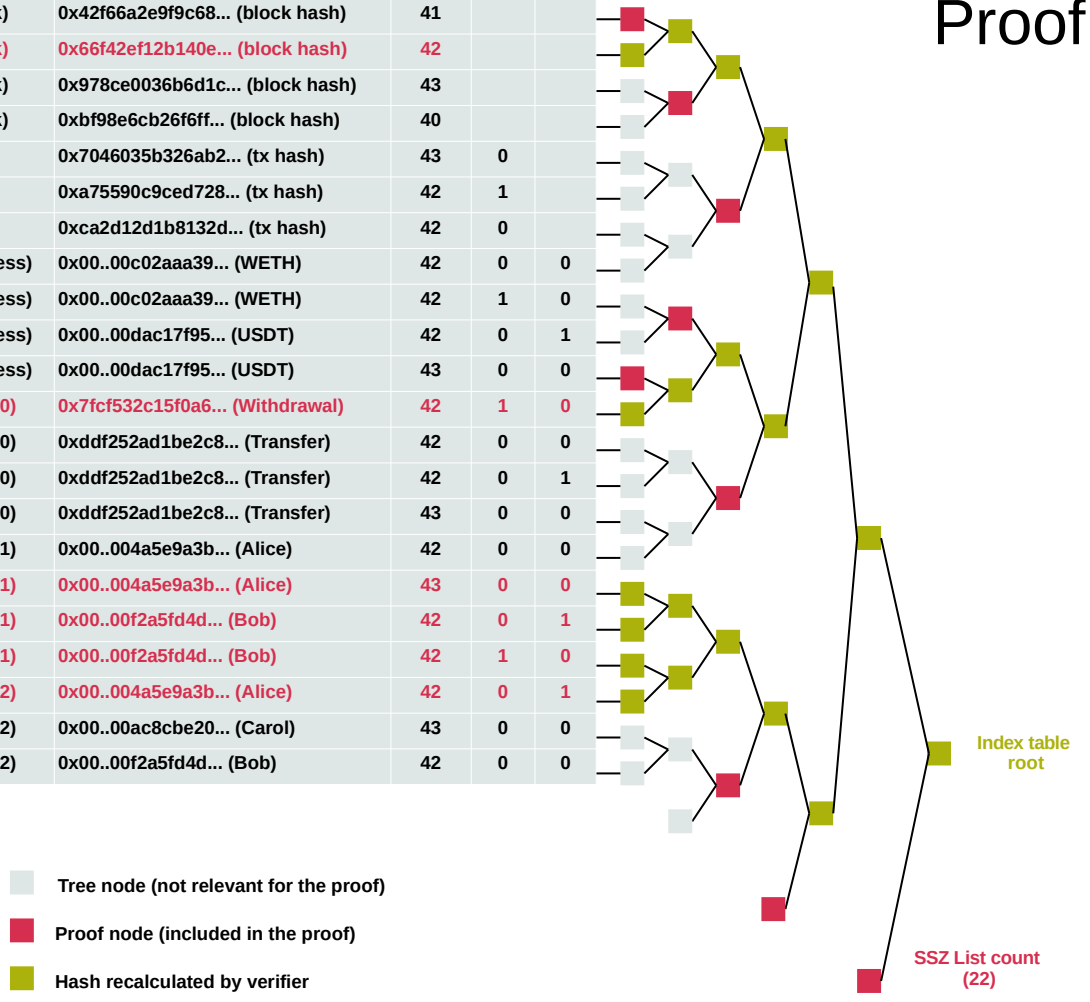
- event type: 1 (transaction hash)
- index value: 0xdeadbeefcafe...

Results: not found

- Prover looks up the searched entry in the table using binary search
- Searched entry does not exist, finds two adjacent entries
- Creates a multi-proof proving those two entries
- Proof cost per index table: two entries (cca 40-45 bytes per entry with compact encoding) plus *ceil(log2(count))* 32 byte proof nodes

Proof example: log pattern search

Type	Value	Block #	Tx #	Log #
0 (Block)	0x42f66a2e9f9c68... (block hash)	41		
0 (Block)	0x66f42ef12b140e... (block hash)	42		
0 (Block)	0x978ce0036b6d1c... (block hash)	43		
0 (Block)	0xbf98e6cb26f6ff... (block hash)	40		
1 (Tx)	0x7046035b326ab2... (tx hash)	43	0	
1 (Tx)	0xa75590c9ced728... (tx hash)	42	1	
1 (Tx)	0xca2d12d1b8132d... (tx hash)	42	0	
2 (Address)	0x00..00c02aaa39... (WETH)	42	0	0
2 (Address)	0x00..00c02aaa39... (WETH)	42	1	0
2 (Address)	0x00..00dac17f95... (USDT)	42	0	1
2 (Address)	0x00..00dac17f95... (USDT)	43	0	0
3 (Topic0)	0x7fcf532c15f0a6... (Withdrawal)	42	1	0
3 (Topic0)	0xddf252ad1be2c8... (Transfer)	42	0	0
3 (Topic0)	0xddf252ad1be2c8... (Transfer)	42	0	1
3 (Topic0)	0xddf252ad1be2c8... (Transfer)	43	0	0
4 (Topic1)	0x00..004a5e9a3b... (Alice)	42	0	0
4 (Topic1)	0x00..004a5e9a3b... (Alice)	43	0	0
4 (Topic1)	0x00..00f2a5fd4d... (Bob)	42	0	1
4 (Topic1)	0x00..00f2a5fd4d... (Bob)	42	1	0
5 (Topic2)	0x00..004a5e9a3b... (Alice)	42	0	1
5 (Topic2)	0x00..00ac8cbe20... (Carol)	43	0	0
5 (Topic2)	0x00..00f2a5fd4d... (Bob)	42	0	0



Block range: #40..#43

Address: 0x00..00dac17f95... (USDT)

Topic0: 0xddf252ad1be2c8... (Transfer)

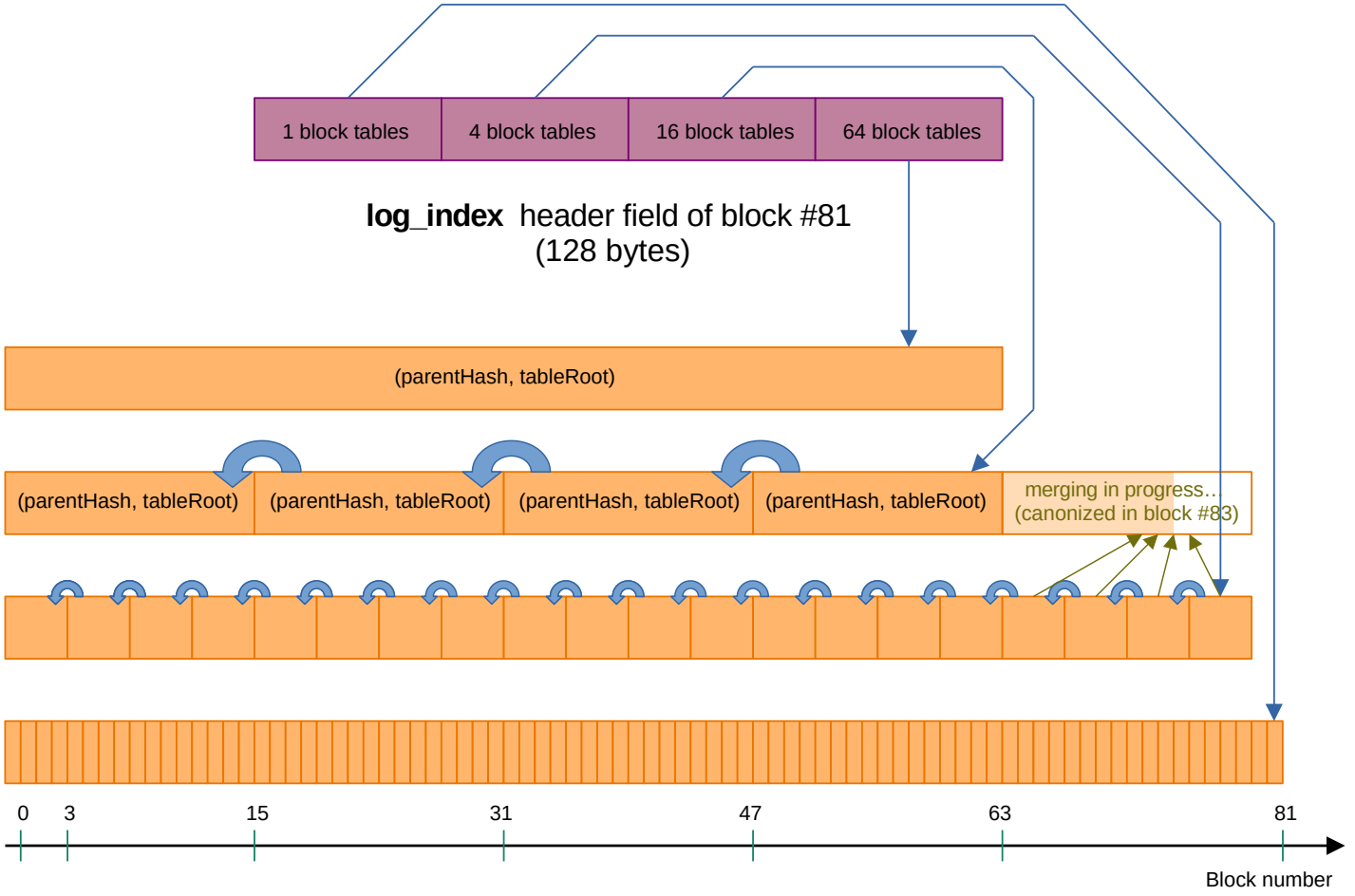
Topic1: 0x00..00f2a5fd4d... (Bob)

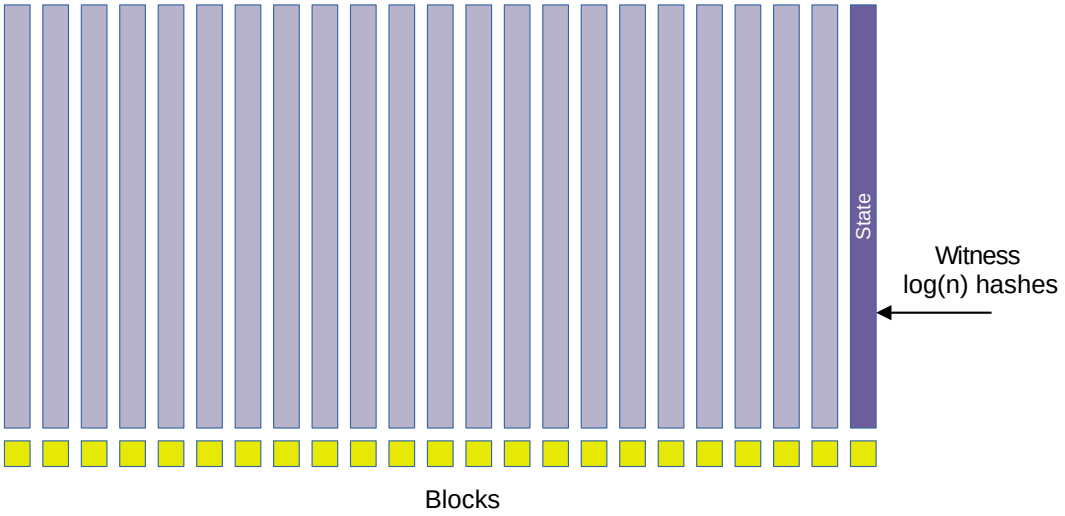
Topic2: anything

Results: [{*Block #42 Tx #0 Log #1*}]

- All occurences of **Topic1 = Bob** are proven:
  - *{Block #42 Tx #0 Log #1}*
  - *{Block #42 Tx #1 Log #0}*
- **Topic0 = Withdrawal** in *{Block #42 Tx #1 Log #0} (no match)*
- *Block #42* root hash is proven
  - Receipt proves that *{Block #42 Tx #0 Log #1} is a match*

Covering the chain history with table chains





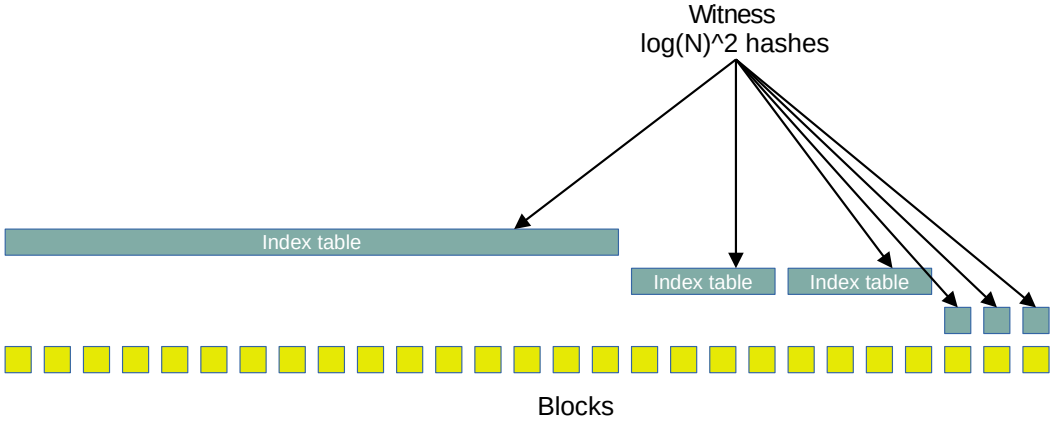
Logs vs State

Pro state tree:

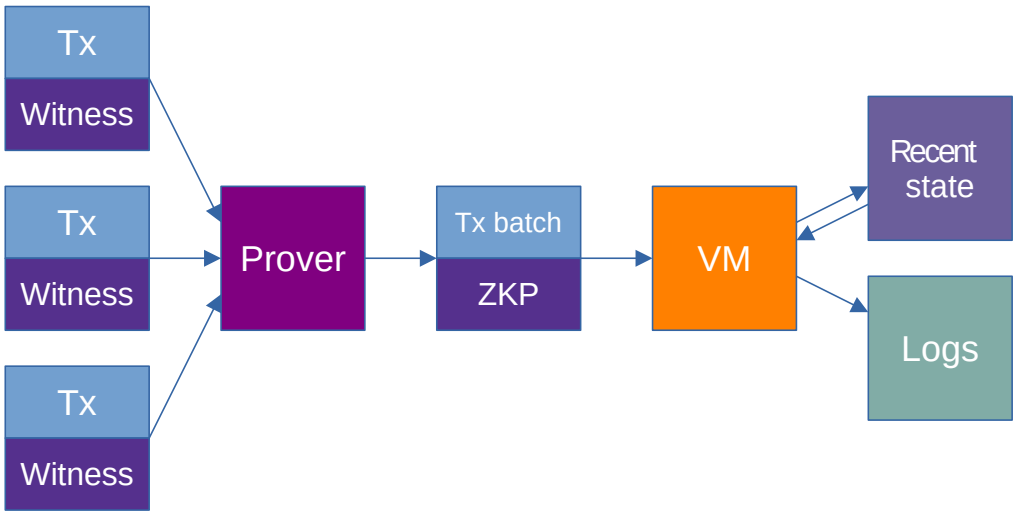
- Smaller witness size

Pro index tables:

- Very cheap to add new entries
- Most of the processing is async/off-chain
- Tables are not changed, only merged (big ones are unchanged for a long time)
- Easy to distribute, no witness provider centralization issue
- Time information is available



Off-chain authorization, pre-checks and witness processing

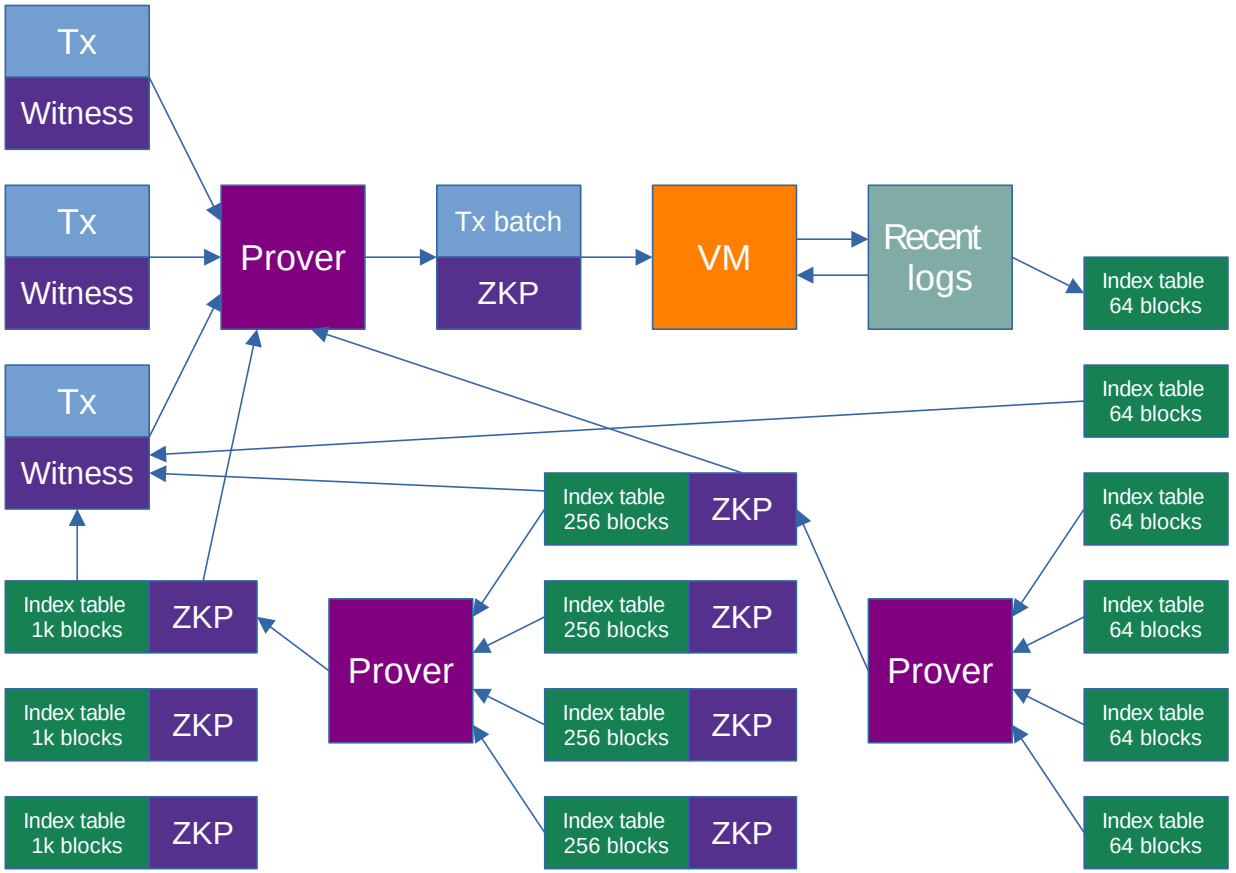


- Witnesses, signatures and other pre-check data can be processed in a ZKP
- Proving cost settled with provers on private channels, on-chain proof verification cost only
- Cheap pre-checks good for
  - Full or partial statelessness
  - Account abstraction, big PQ signatures

Logs as contract storage

- Implement the Trustless Log Index (EIP-7745B, much simpler than original EIP-7745)
  - Very cheap to append, easy to expire, easy to distribute
  - Big index tables merged with ZK proofs (immutable once max table size reached)
  - Add erasure coding, ensure data availability until tables expired
  - Keeping relevant witnesses from expired tables is the user’s responsibility
- Make logs accessible from the EVM
  - Recent ones without witness, older ones with witness
- Can simulate state but more flexible
- Append-only, content/time addressed storage
  - Expiry can be controlled by contract (only search non-expired index range)
- Parallel execution friendly (sets, lists, tables without write collisions)
- Multi-chain friendly (easy to list relevant events since last update)

Prover infrastructure of a log-based storage architecture



Thank you for your attention!

Trustless Log Index repository:

https://github.com/zsfelfoldi/tli

Contact:

zsfelfoldi @ discord

zsfelfoldi@ethereum.org