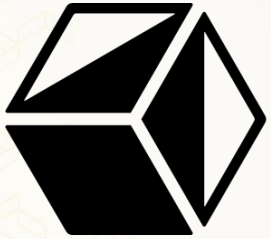
How Hardhat 3 will ensure precise simulation for L2s using EDR

Wodann

Hardhat



NOMIC
FOUNDATION

Hardhat

Agenda

1. What is EDR?
2. Variability between L2s
3. Problems when developing for L2s
4. How does EDR simulate L2s?
5. L2s in Hardhat 3

01 | What is EDR?

What is EDR?

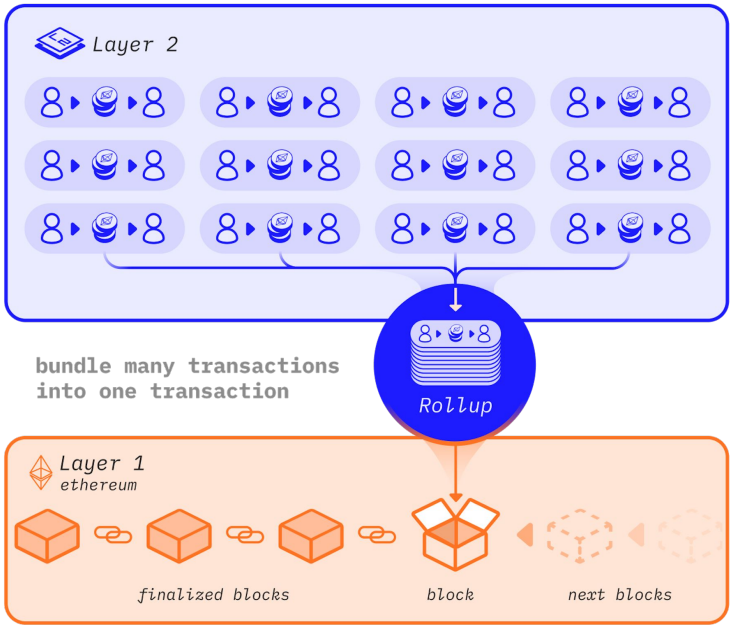
- Ethereum Development Runtime
- EVM development runtime library for tooling
  - Blockchain simulation
  - Observing EVM / Solidity execution
- Targeting smart contract development
  - Simulation, testing & debugging
  - Not targeting to be an execution layer (EL) node

EDR Launch Announcement

02 |

Variability between L2s

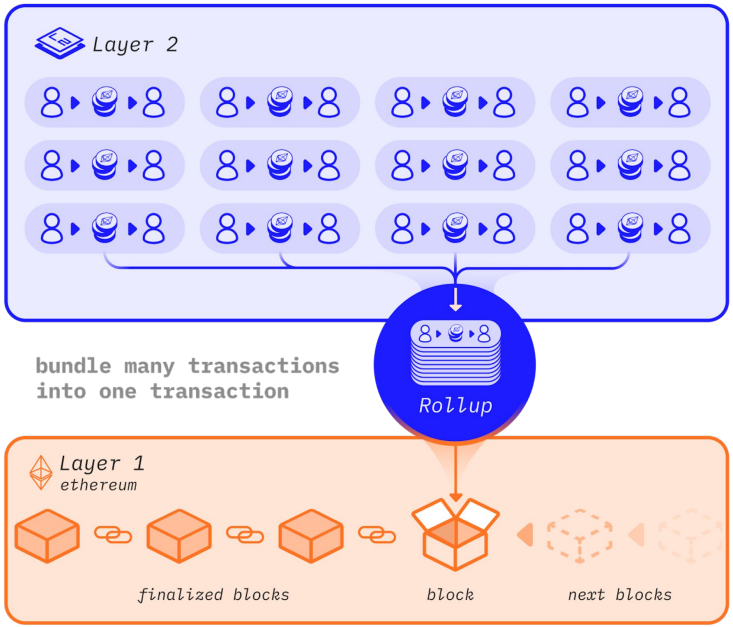
Variability between L2s



Ethereum™ Foundation CC BY 4.0

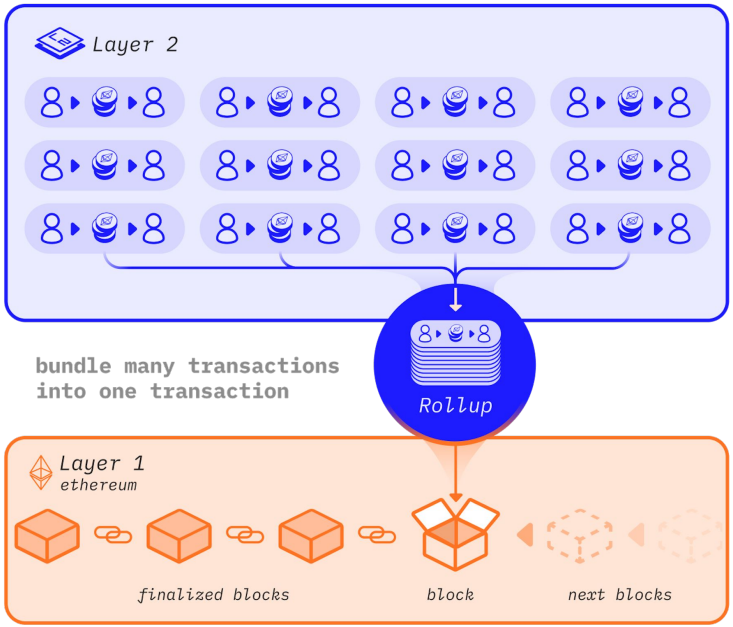
- EVM equivalent L2s
  - Comply with EVM specification

Variability between L2s



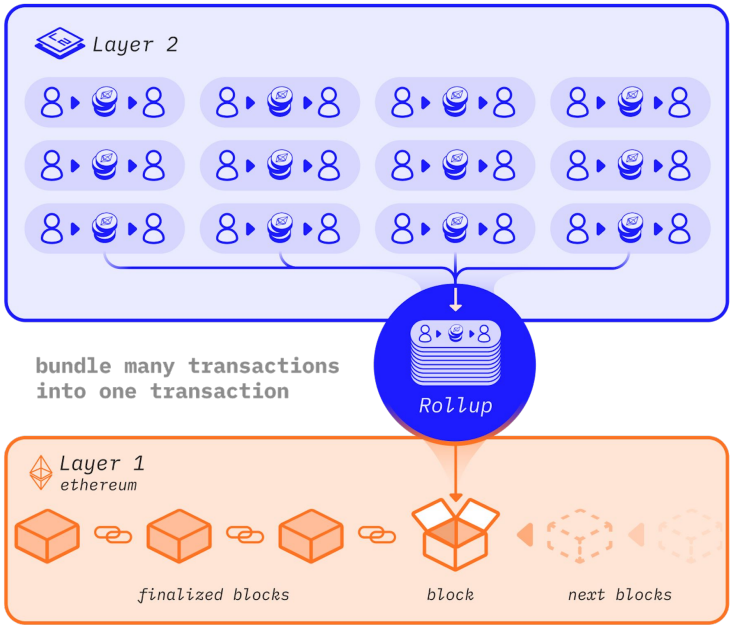
- EVM equivalent L2s
  - Comply with EVM specification
- Transactions
  - Logic
  - Halt Reasons

Variability between L2s



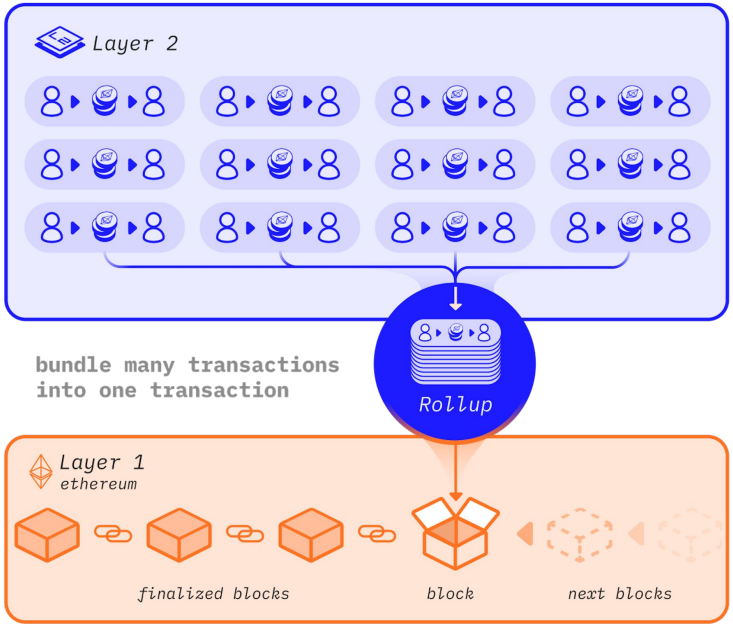
- EVM equivalent L2s
  - Comply with EVM specification
- Transactions
  - Logic
  - Halt Reasons
- Opcodes

Variability between L2s



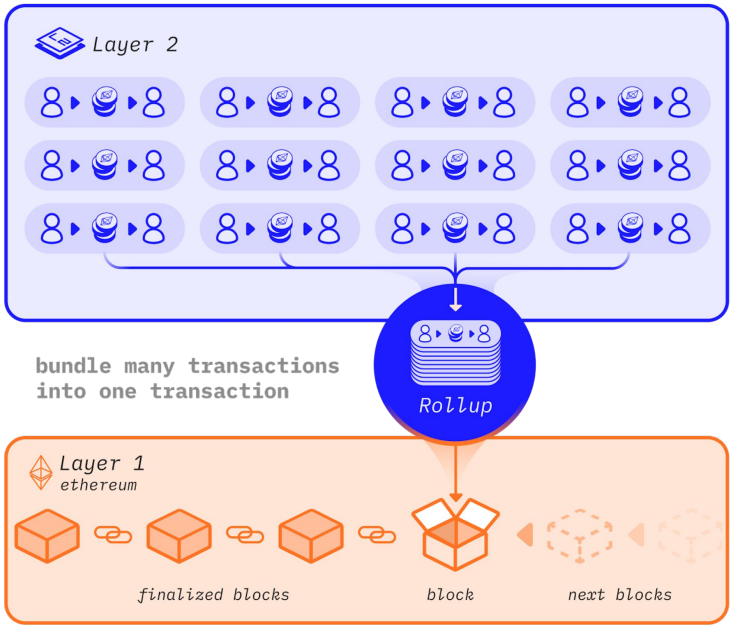
- Precompiles

Variability between L2s



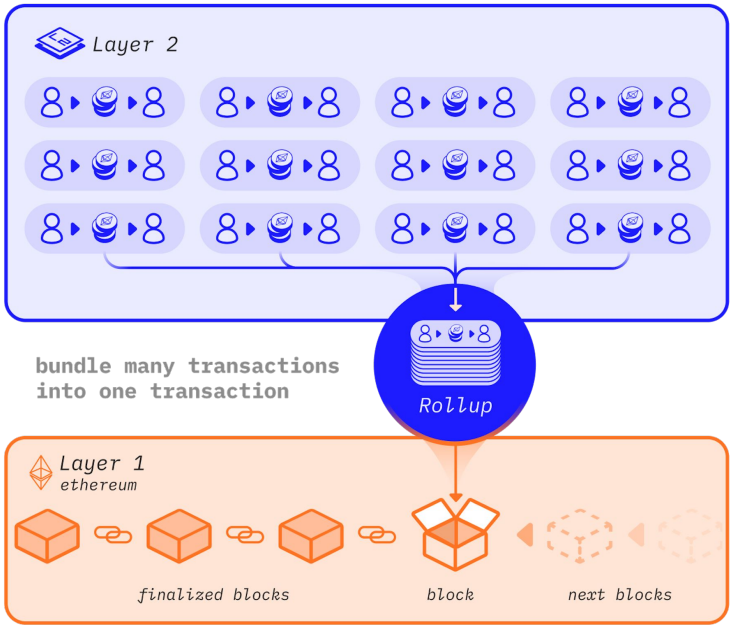
- Precompiles
- Hardforks

Variability between L2s



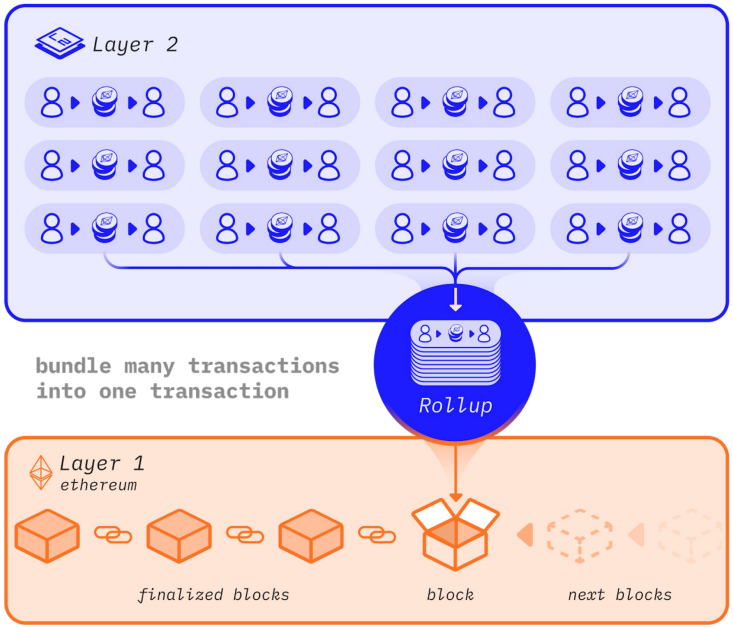
- Precompiles
- Hardforks
- Blocks
  - Fees
  - Transaction receipts

Variability between L2s



- (Pre-)deployed contracts

Variability between L2s



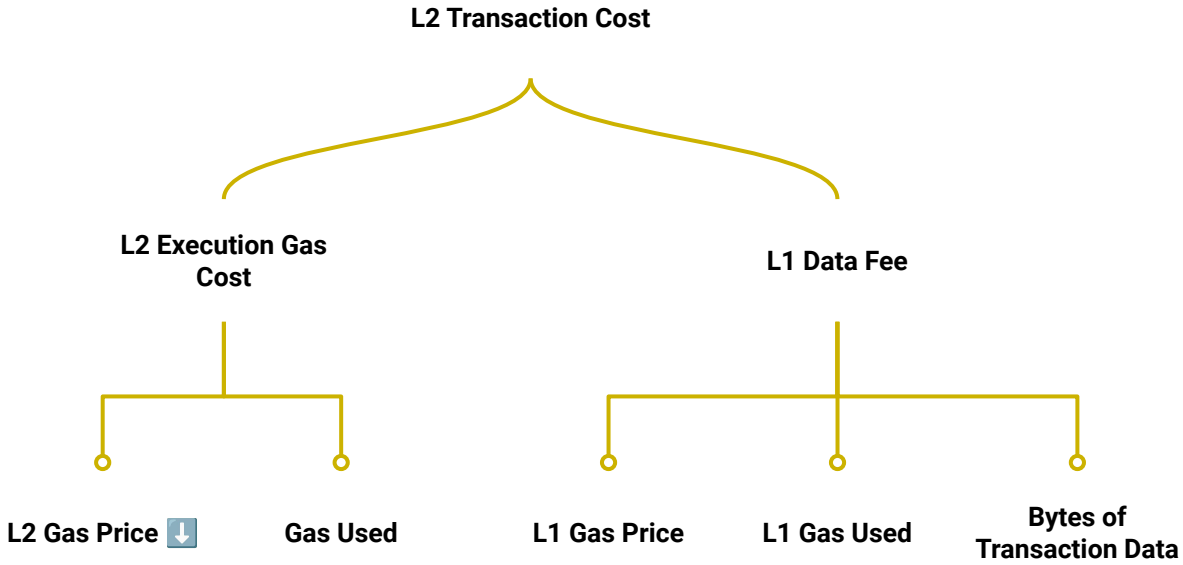
- (Pre-)deployed contracts
- RPC
  - Additional fields
  - Fields with different behavior
  - Methods with different logic

03 | Problems when developing for L2s

Problems when developing for L2s

- Incorrect L2 execution
  - Unknown transaction types
  - Different opcode behavior
- Invalid L2 blocks
  - Invalid RLP encoding
  - Different header fields
- Inaccurate gas calculation

Problems when developing for L2s

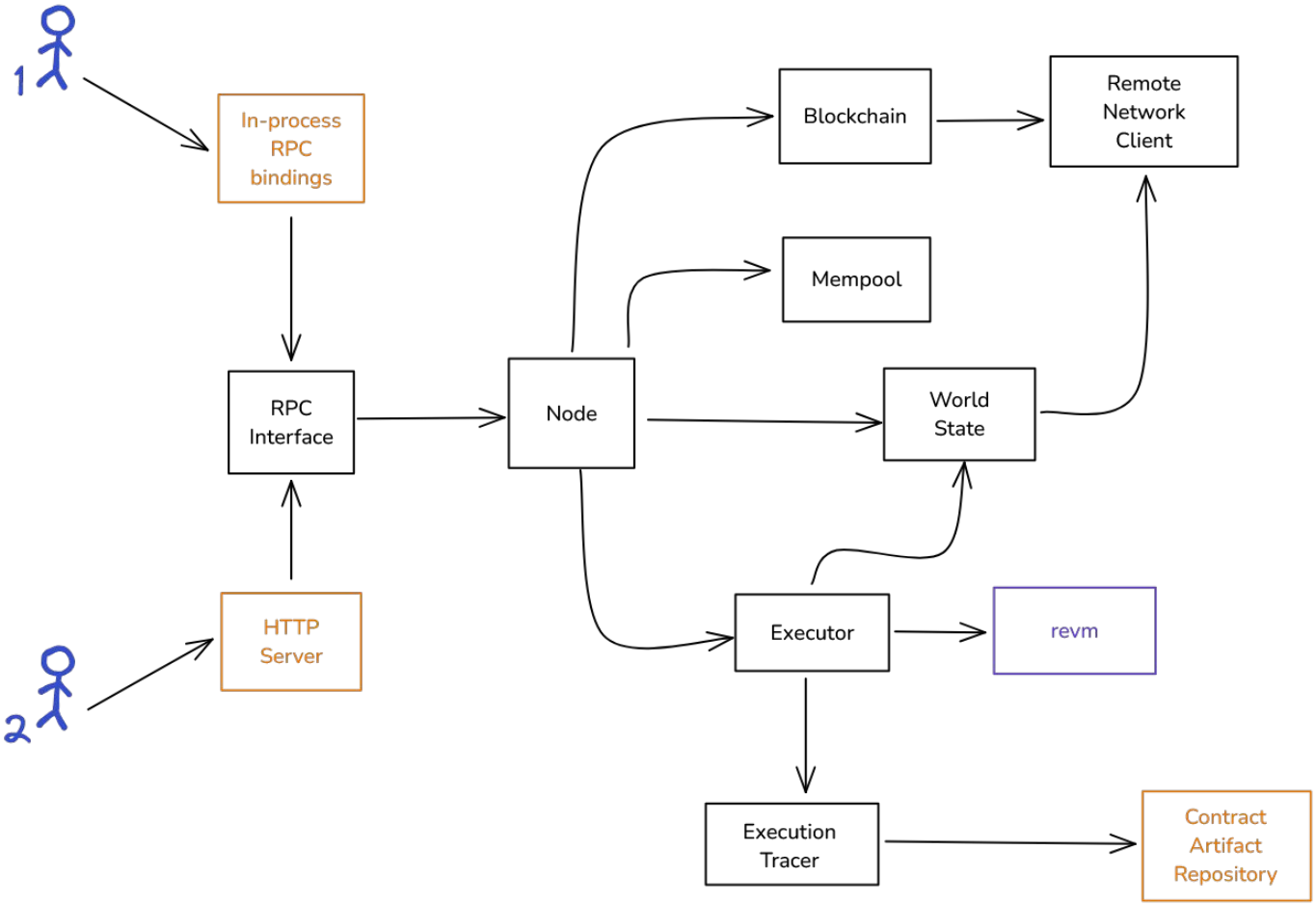


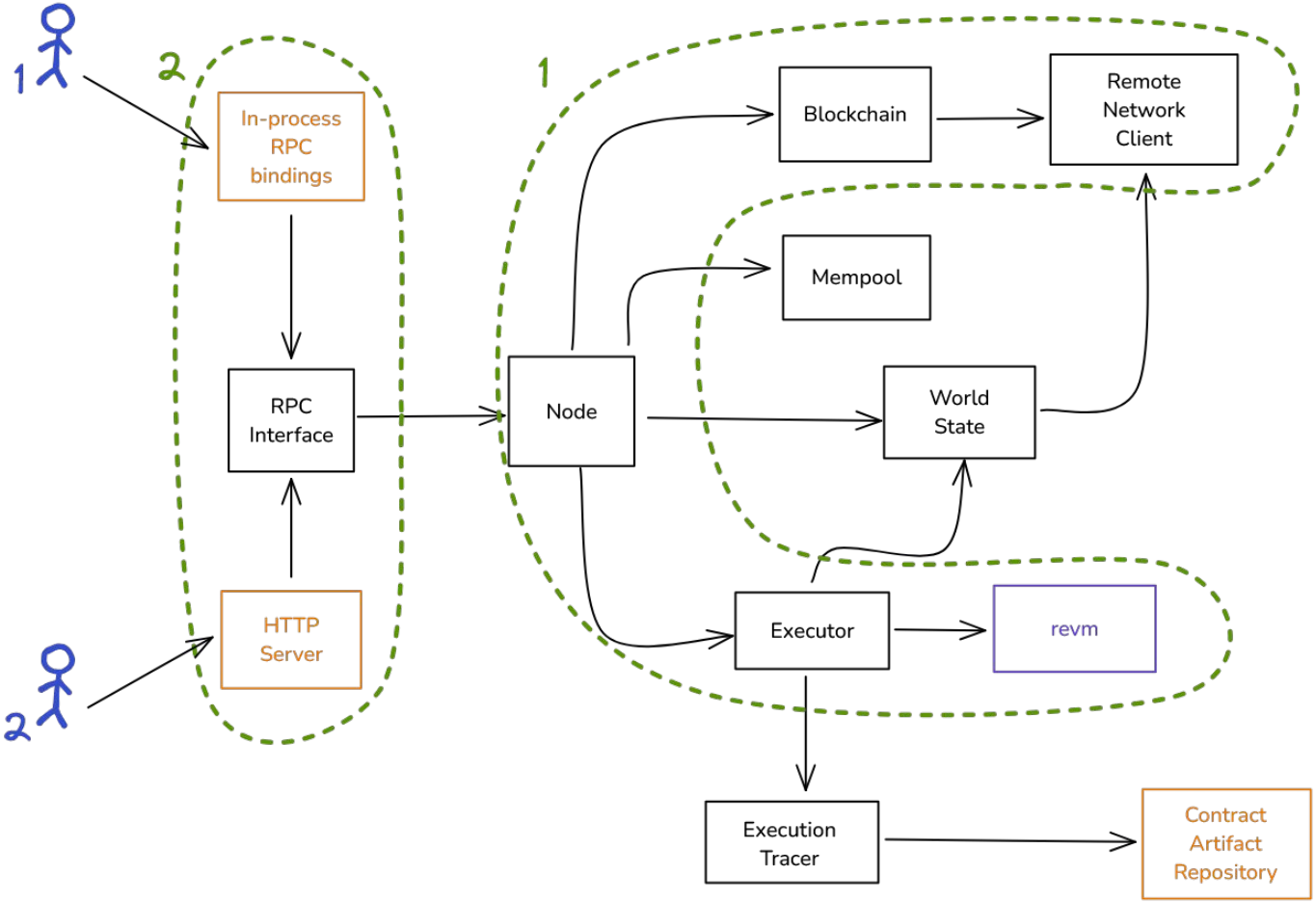
Problems when developing for L2s

- Debugging on deviant state
  - Treat unknown transaction as EIP-155
  - Skip unknown transactions
- Building L2 smart contracts using L1 tools
  - Hoping it works in L2
  - False sense of security
  - Leaves room for security vulnerability

04 |

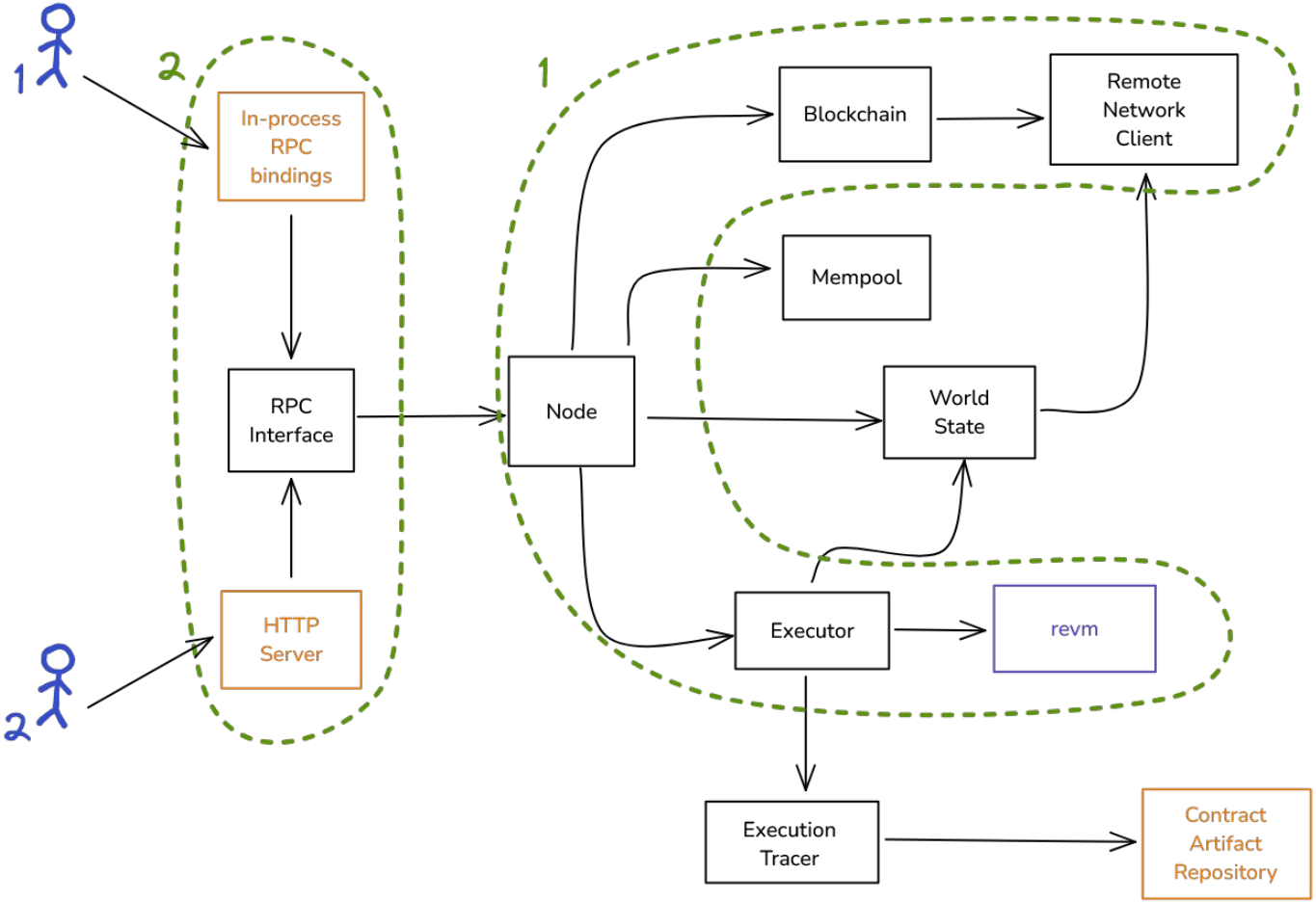
How does EDR simulate L2s?

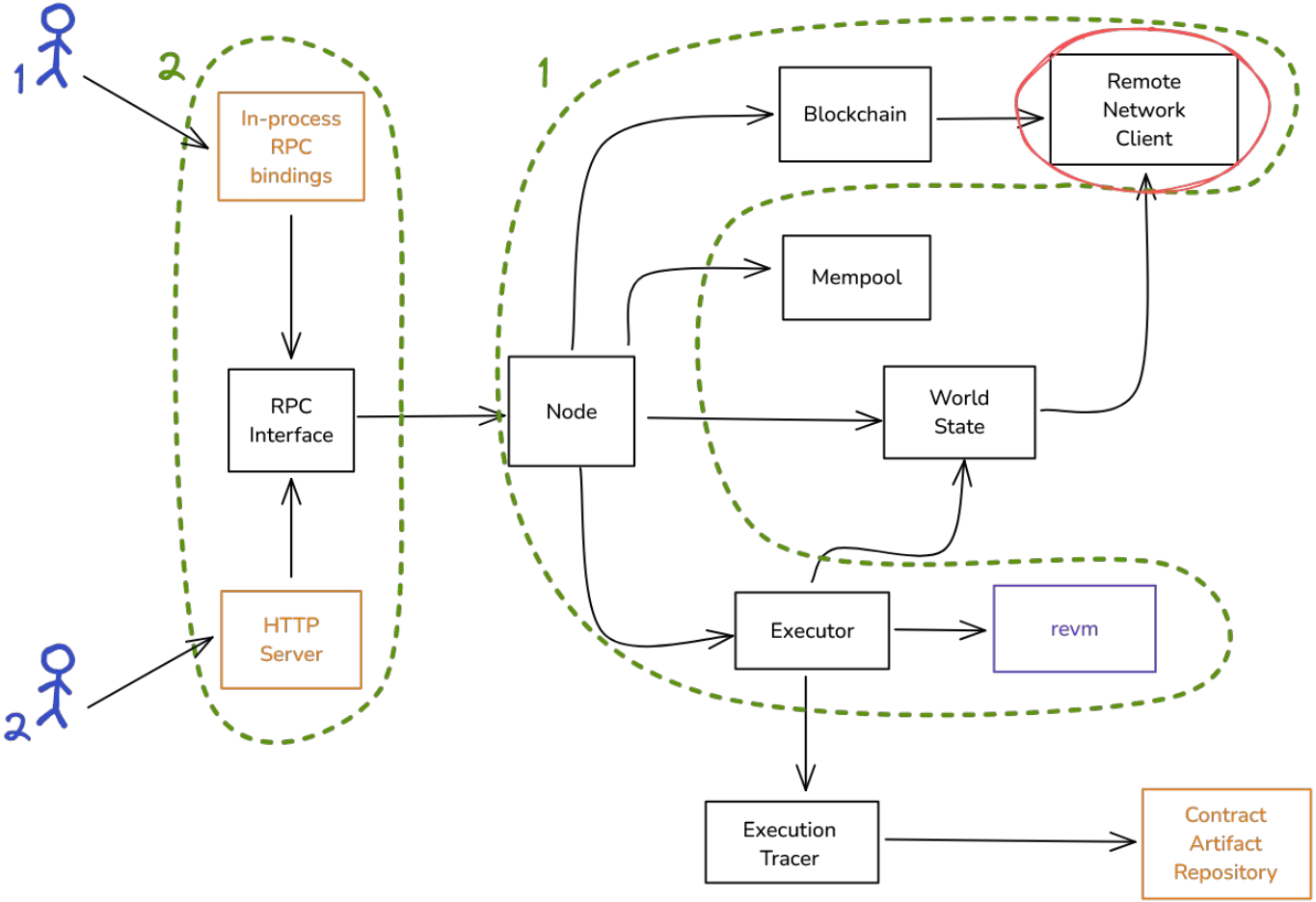

1
In-process RPC bindings
Blockchain
Remote Network Client
Mempool
RPC Interface
Node
World State
HTTP Server
Executor
revm
2
Execution Tracer
Contract Artifact Repository

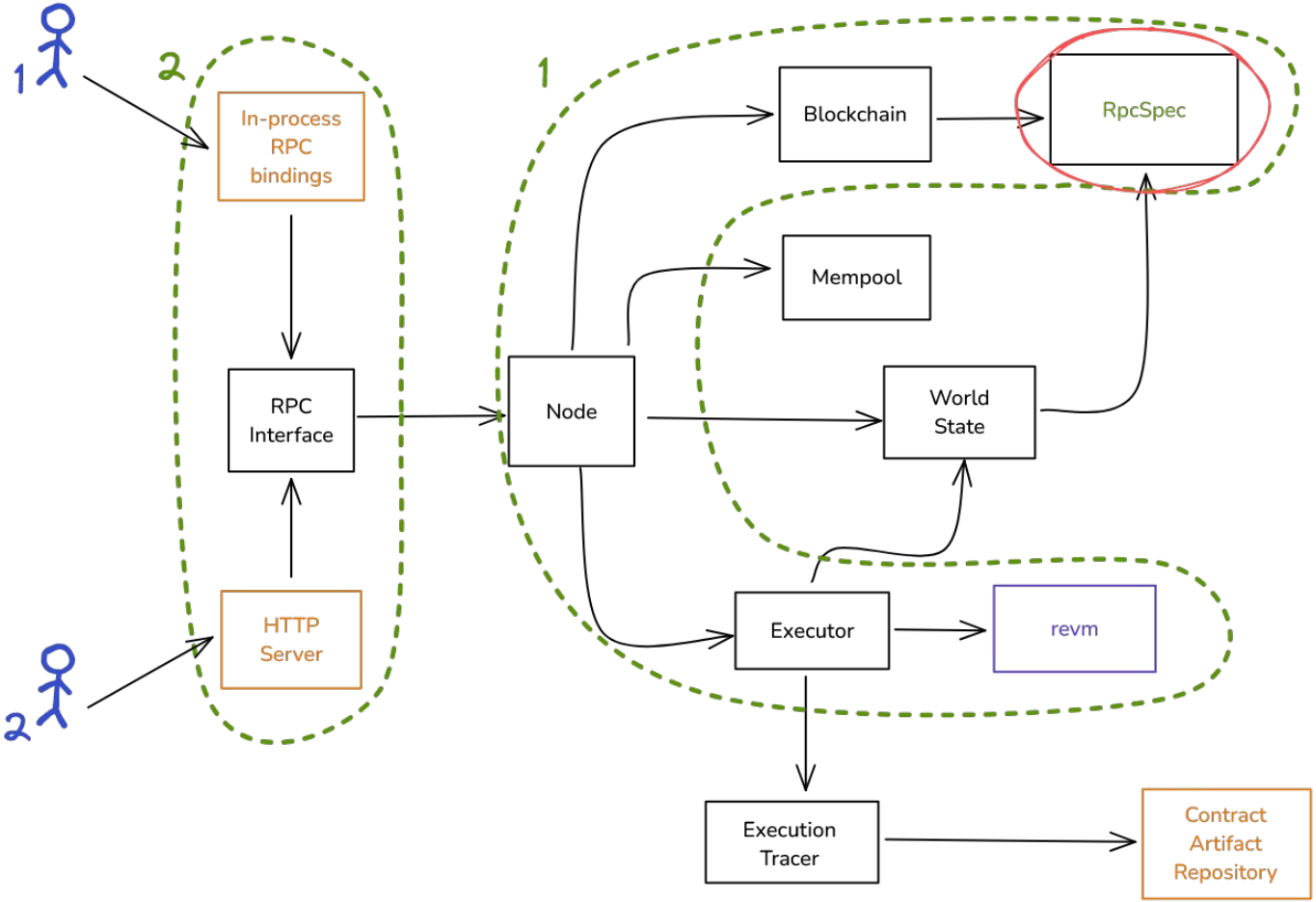

1
2
In-process
RPC
bindings
RPC
Interface
HTTP
Server
2
1
Blockchain
Remote
Network
Client
Mempool
Node
World
State
Executor
revm
Execution
Tracer
Contract
Artifact
Repository

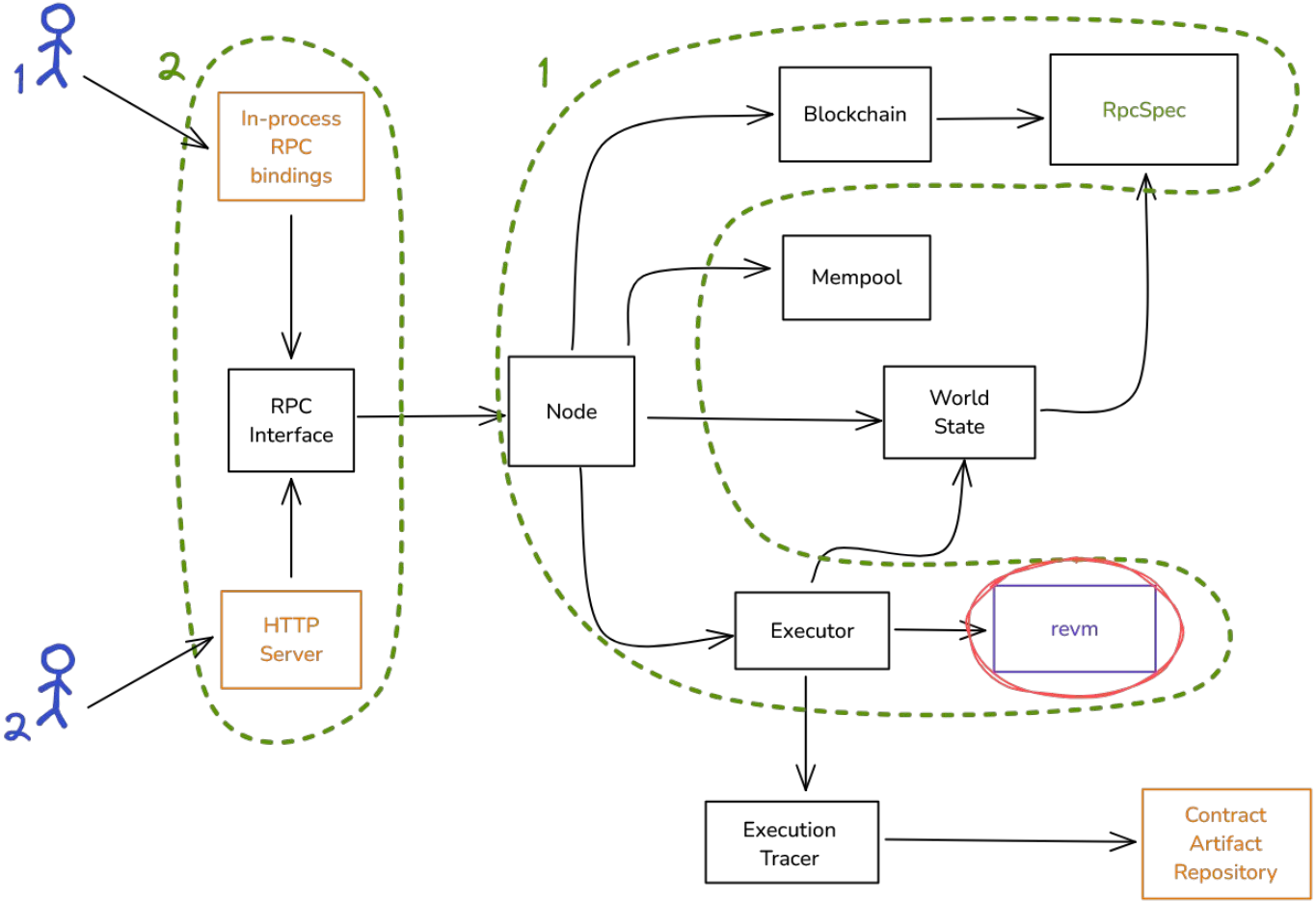
How does EDR simulate L2s?

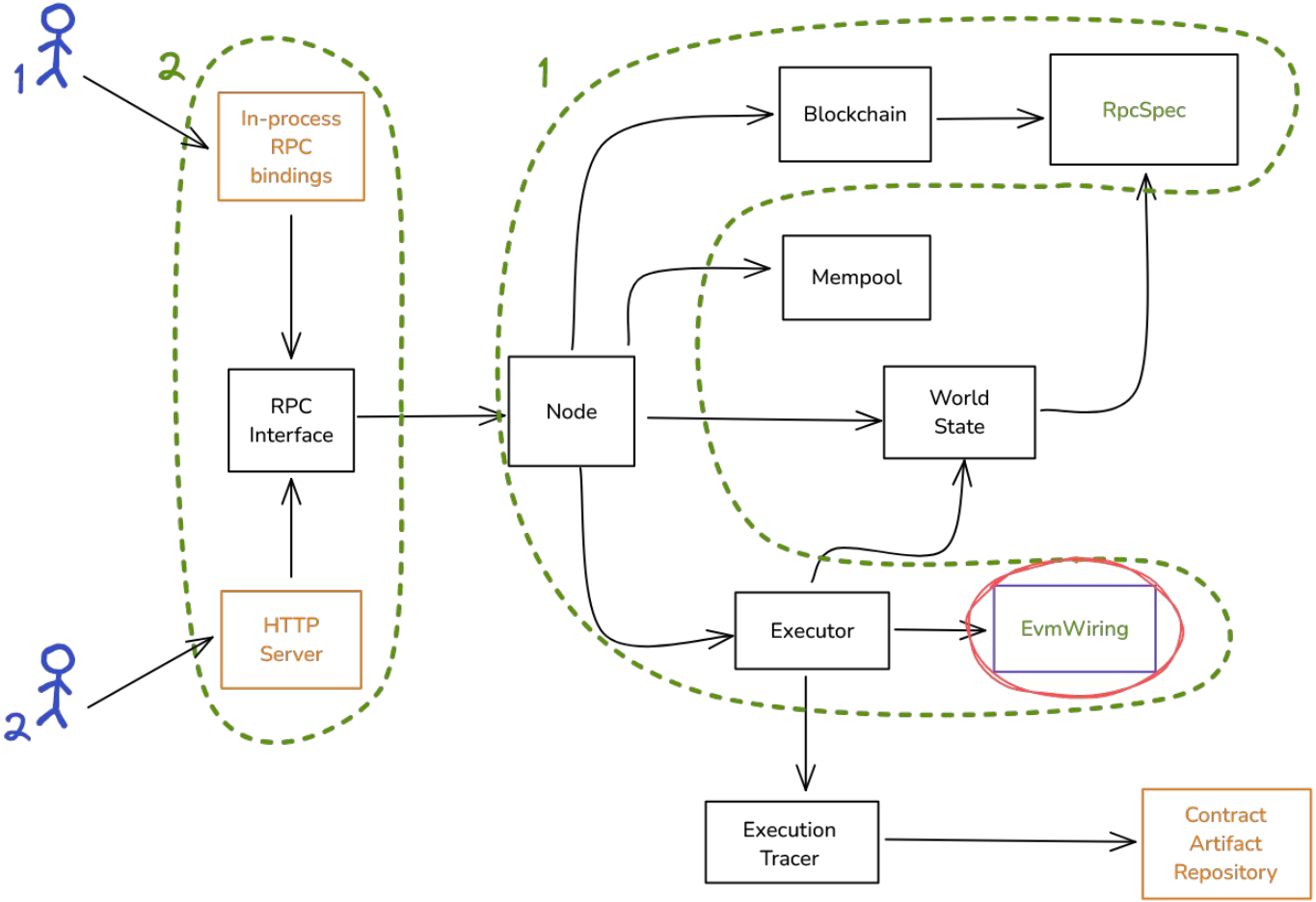
- Extensibility in Rust
  - Compile-time polymorphism
  - Generate type errors at compile-time
  - Reusability of base chain types
- Traits & generics
  - Associated types and constants
  - Distribution using Rust crates

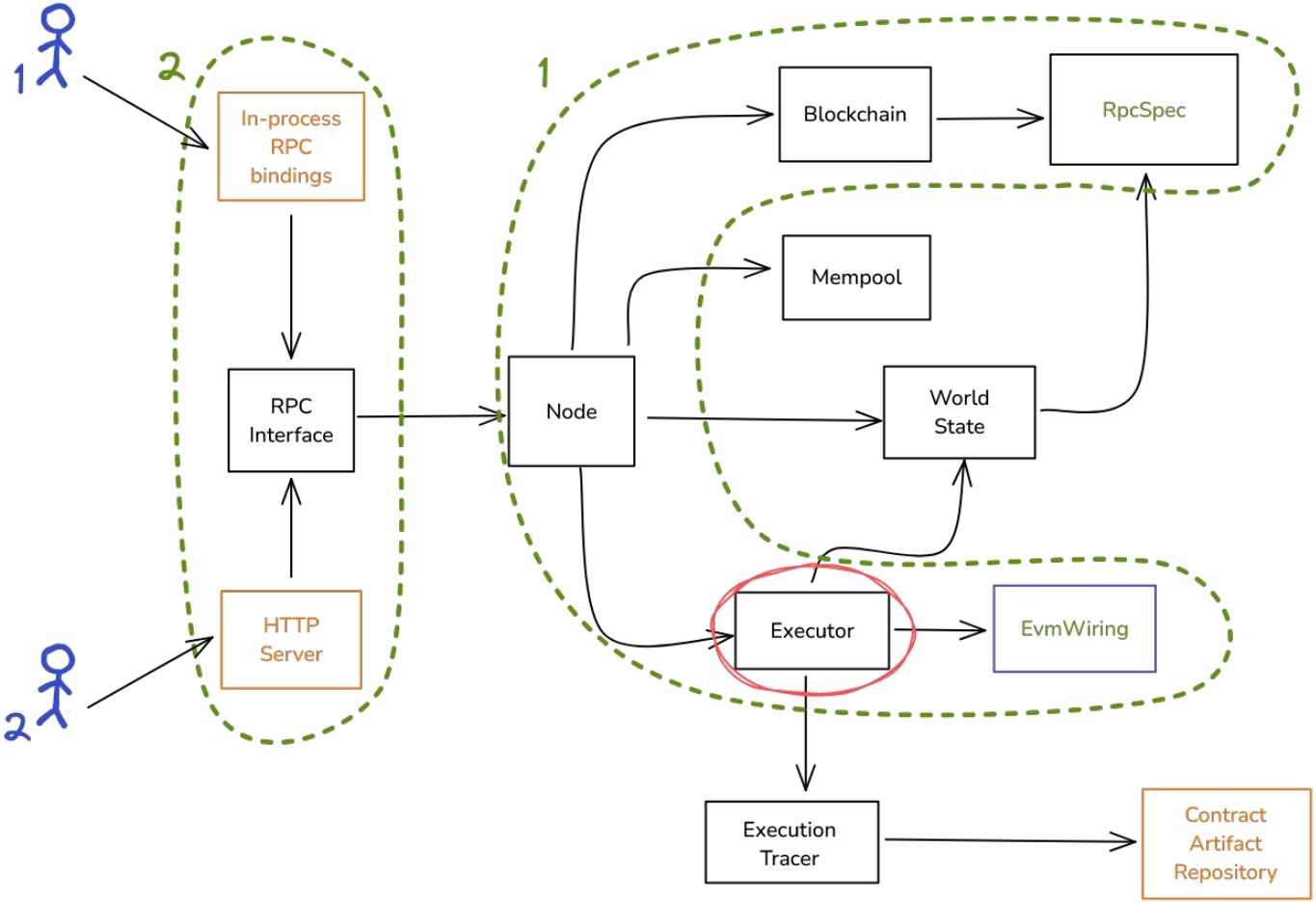

1
2
In-process
RPC
bindings
RPC
Interface
HTTP
Server
2
1
Blockchain
Remote
Network
Client
Mempool
Node
World
State
Executor
revm
Execution
Tracer
Contract
Artifact
Repository

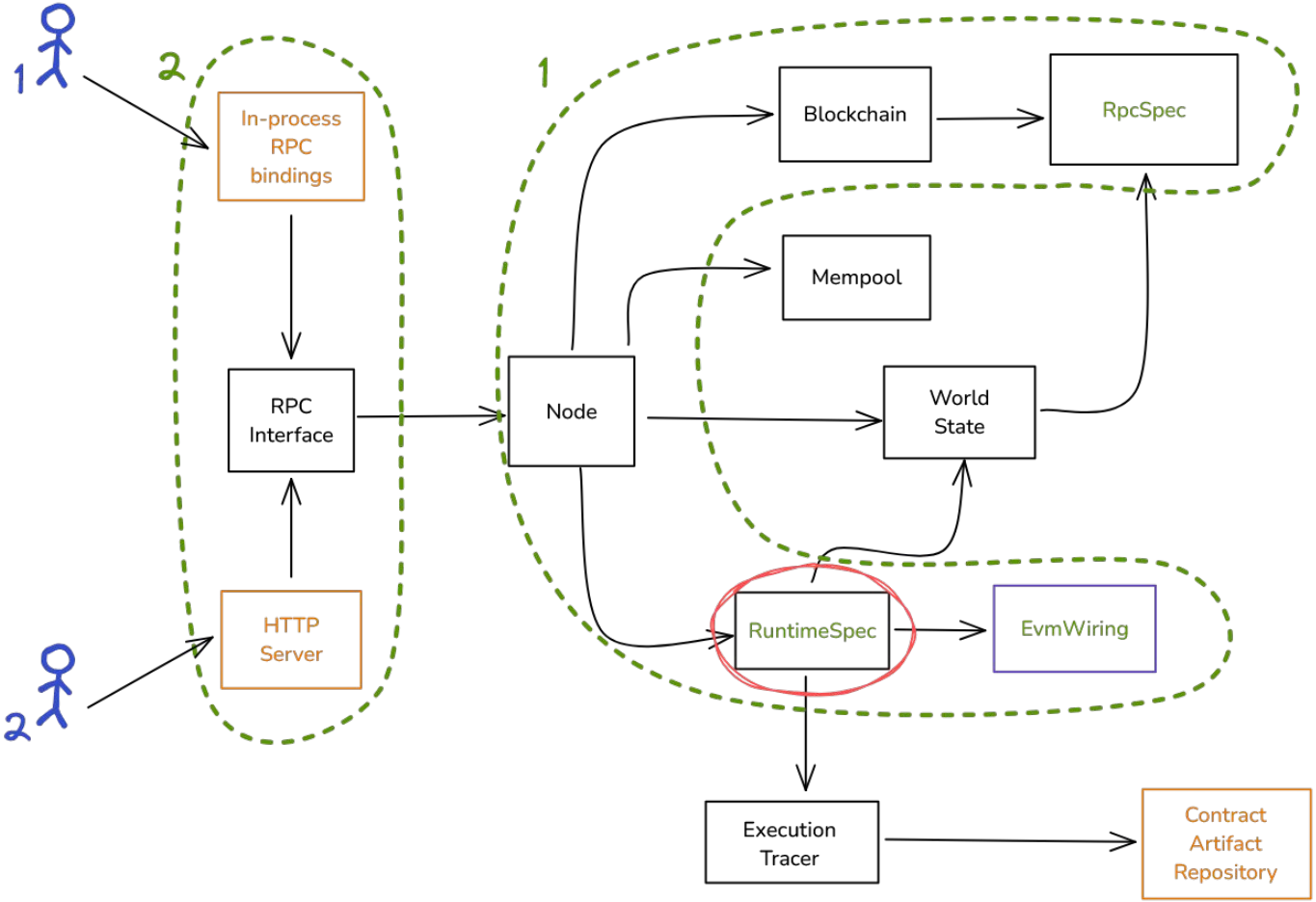

1
2
In-process
RPC
bindings
RPC
Interface
HTTP
Server
2
1
Blockchain
Remote
Network
Client
Mempool
Node
World
State
Executor
revm
Execution
Tracer
Contract
Artifact
Repository

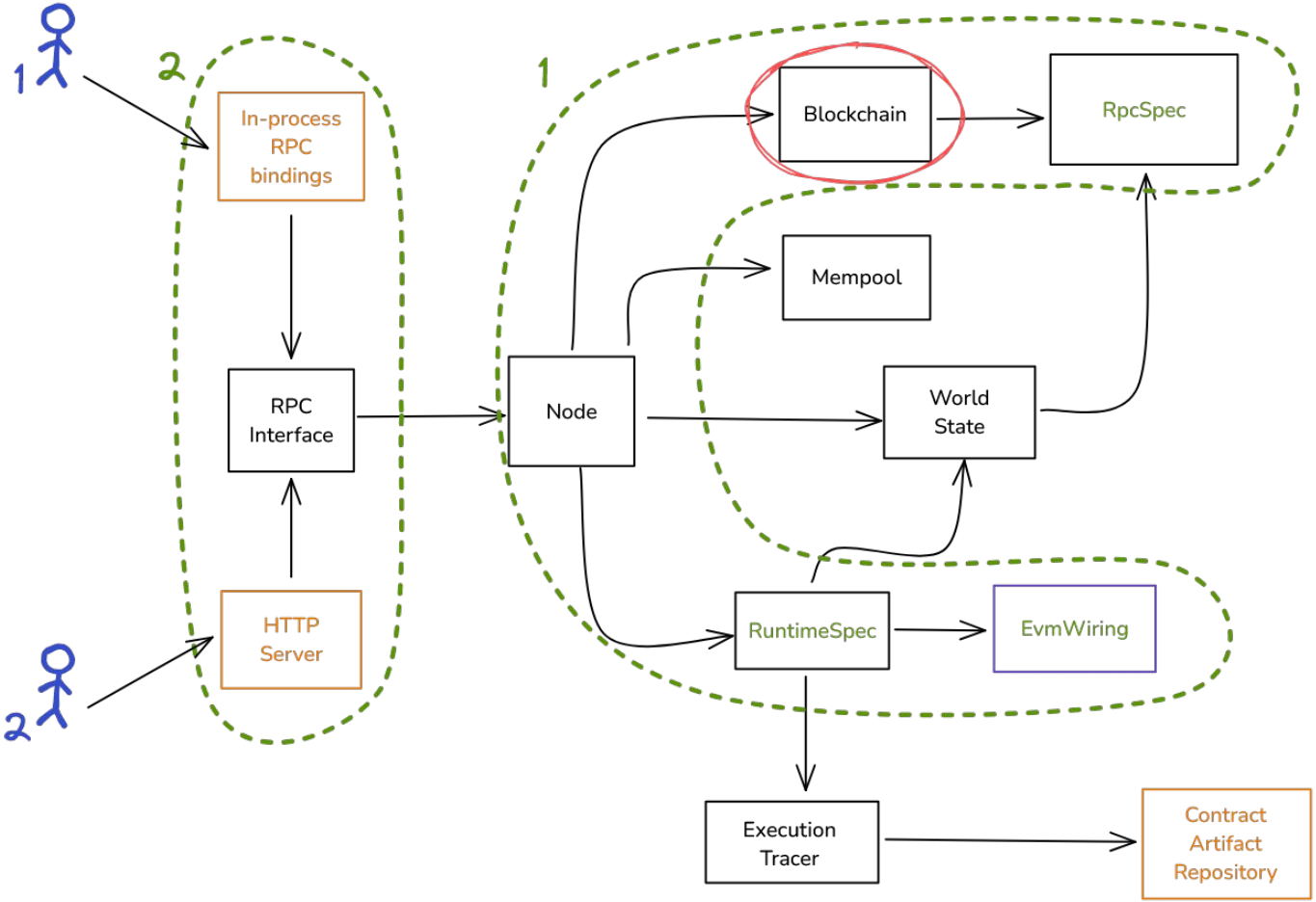

1
2
In-process RPC bindings
RPC Interface
HTTP Server
2
1
Blockchain
RpcSpec
Mempool
Node
World State
Executor
revm
Execution Tracer
Contract Artifact Repository

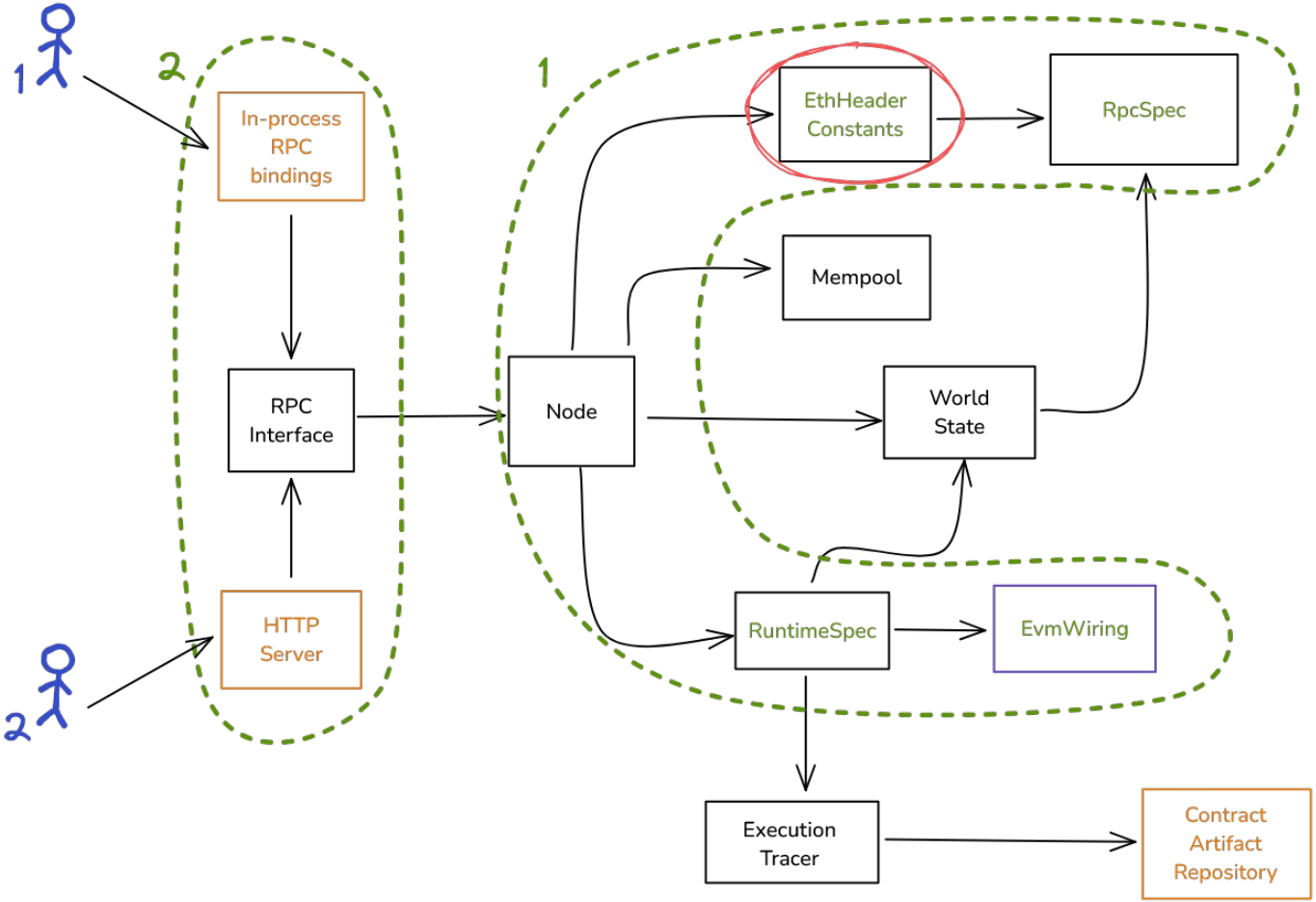

1
2
In-process RPC bindings
RPC Interface
HTTP Server
2
1
Blockchain
RpcSpec
Mempool
Node
World State
Executor
revm
Execution Tracer
Contract Artifact Repository

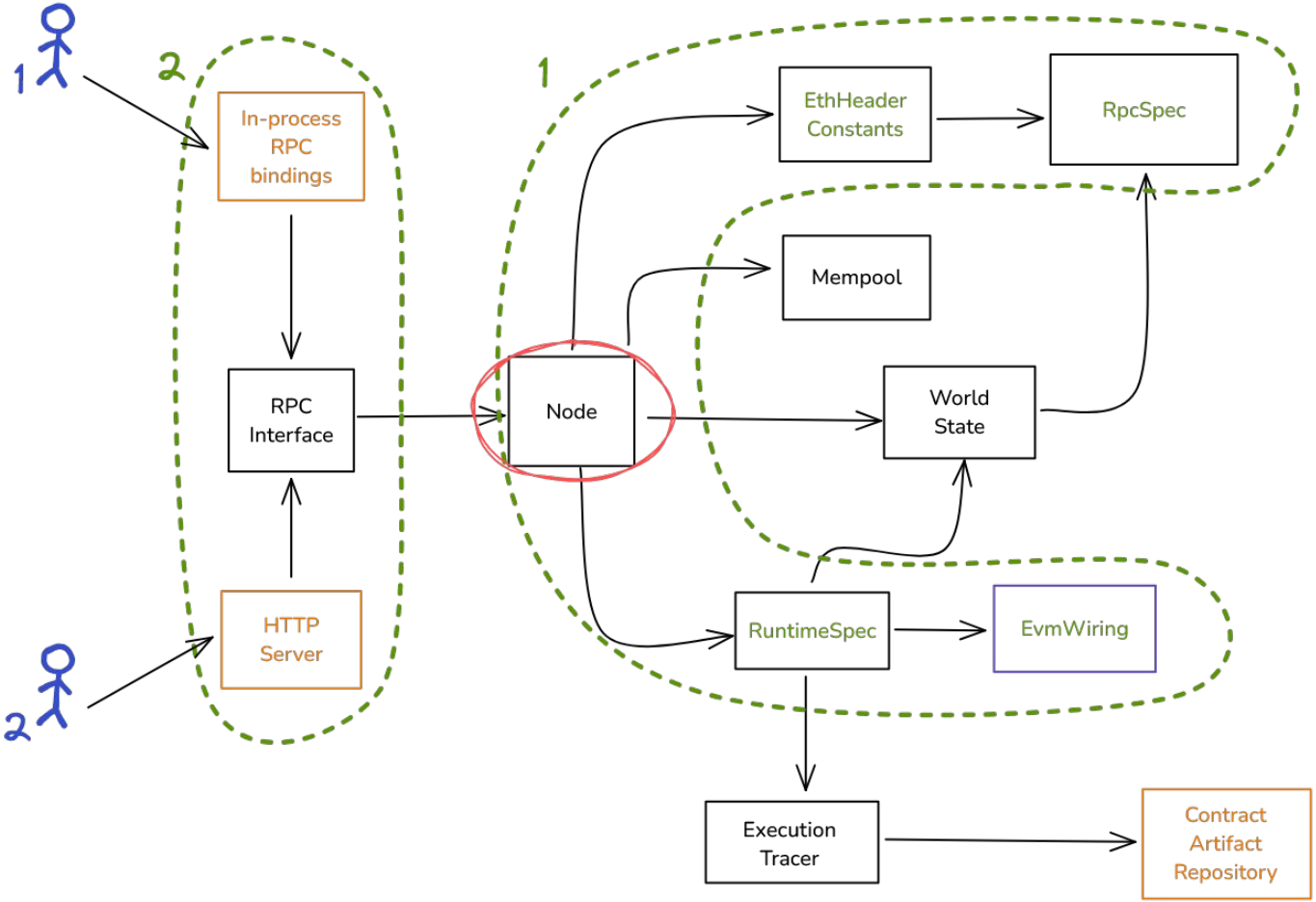

1
2
In-process RPC bindings
RPC Interface
HTTP Server
2
1
Blockchain
RpcSpec
Mempool
Node
World State
Executor
EvmWiring
Execution Tracer
Contract Artifact Repository

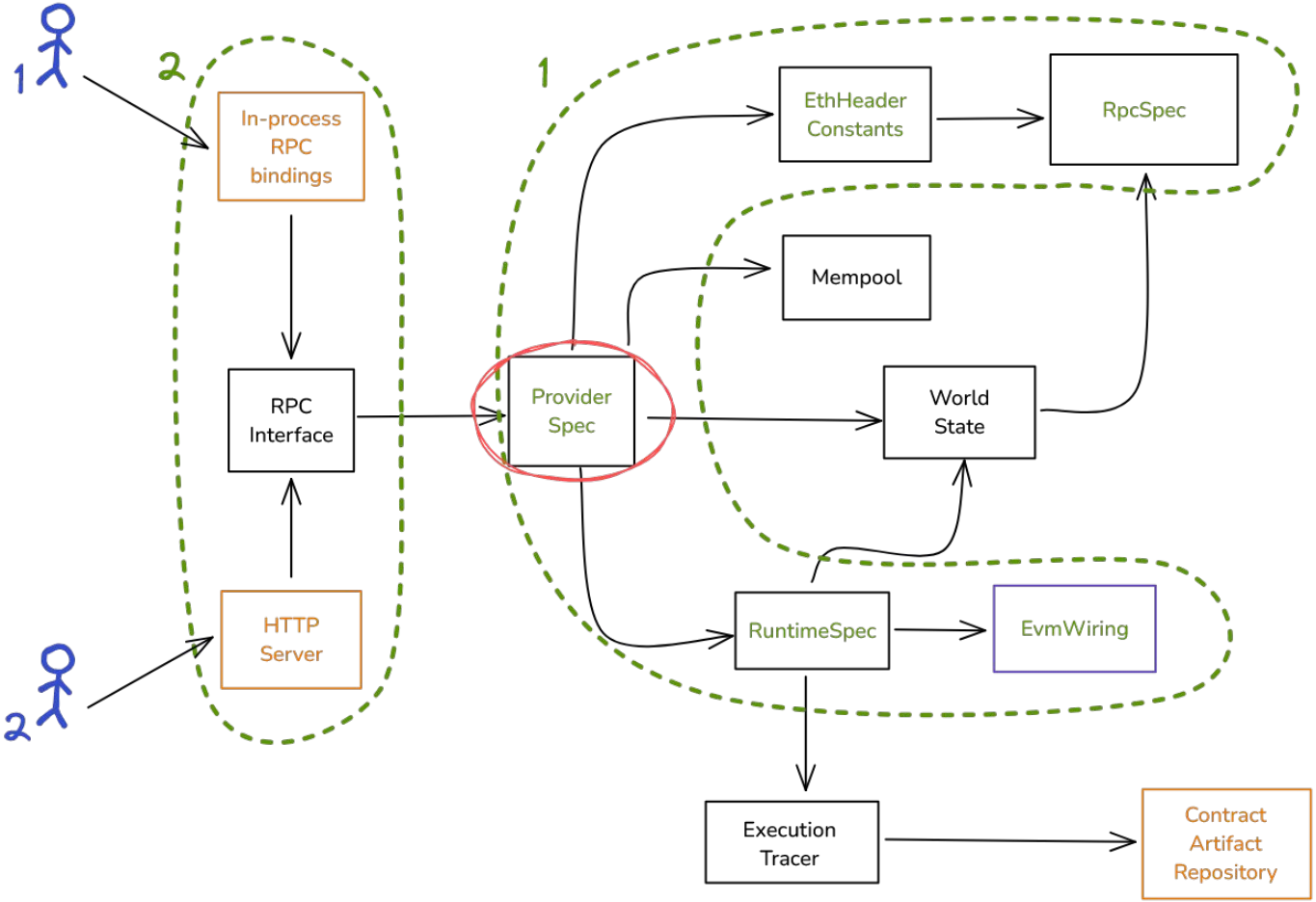

1
2
In-process RPC bindings
RPC Interface
HTTP Server
2
1
Blockchain
RpcSpec
Mempool
Node
World State
Executor
EvmWiring
Execution Tracer
Contract Artifact Repository

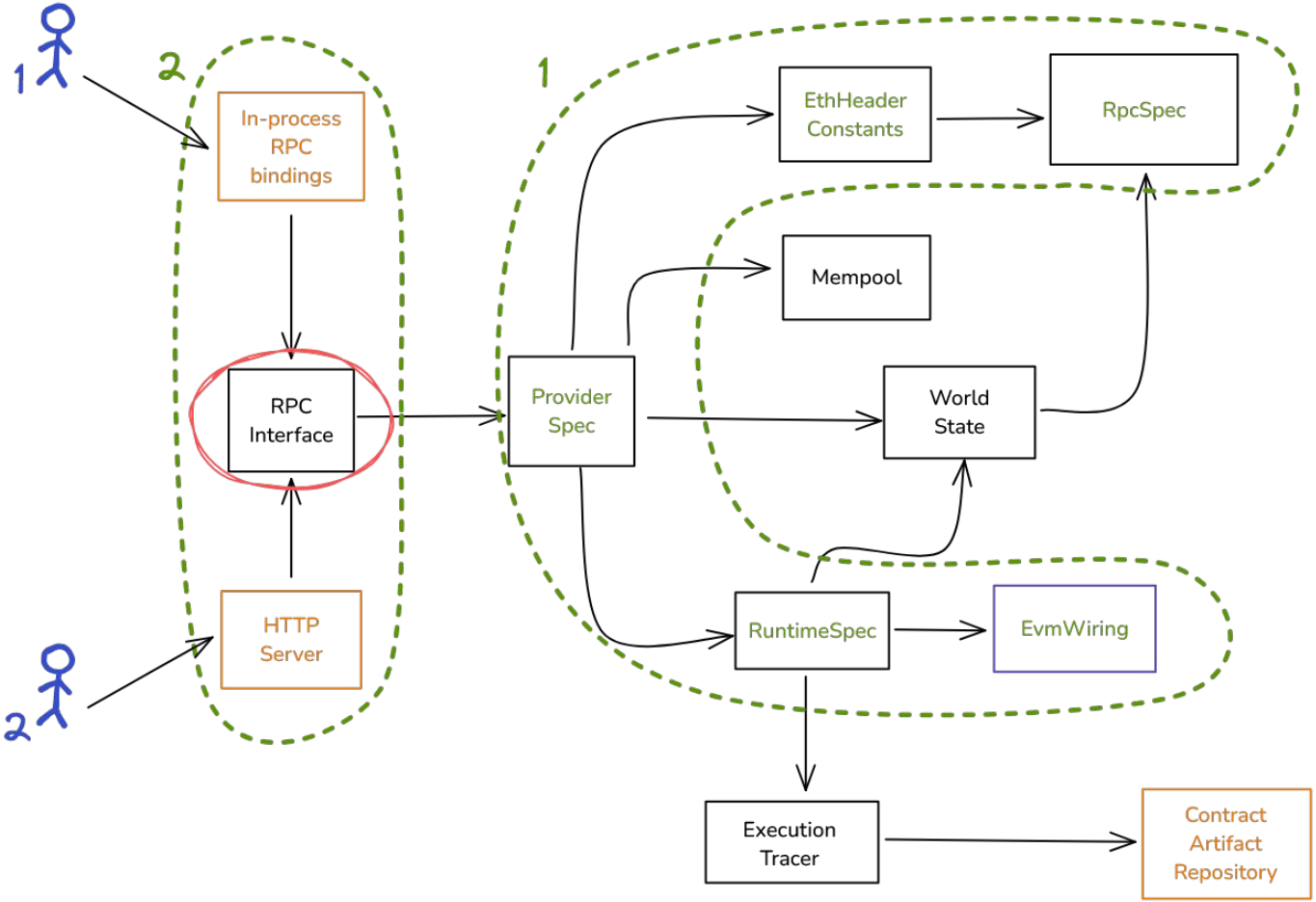

1
2
In-process RPC bindings
RPC Interface
HTTP Server
2
1
Blockchain
RpcSpec
Mempool
Node
World State
RuntimeSpec
EvmWiring
Execution Tracer
Contract Artifact Repository

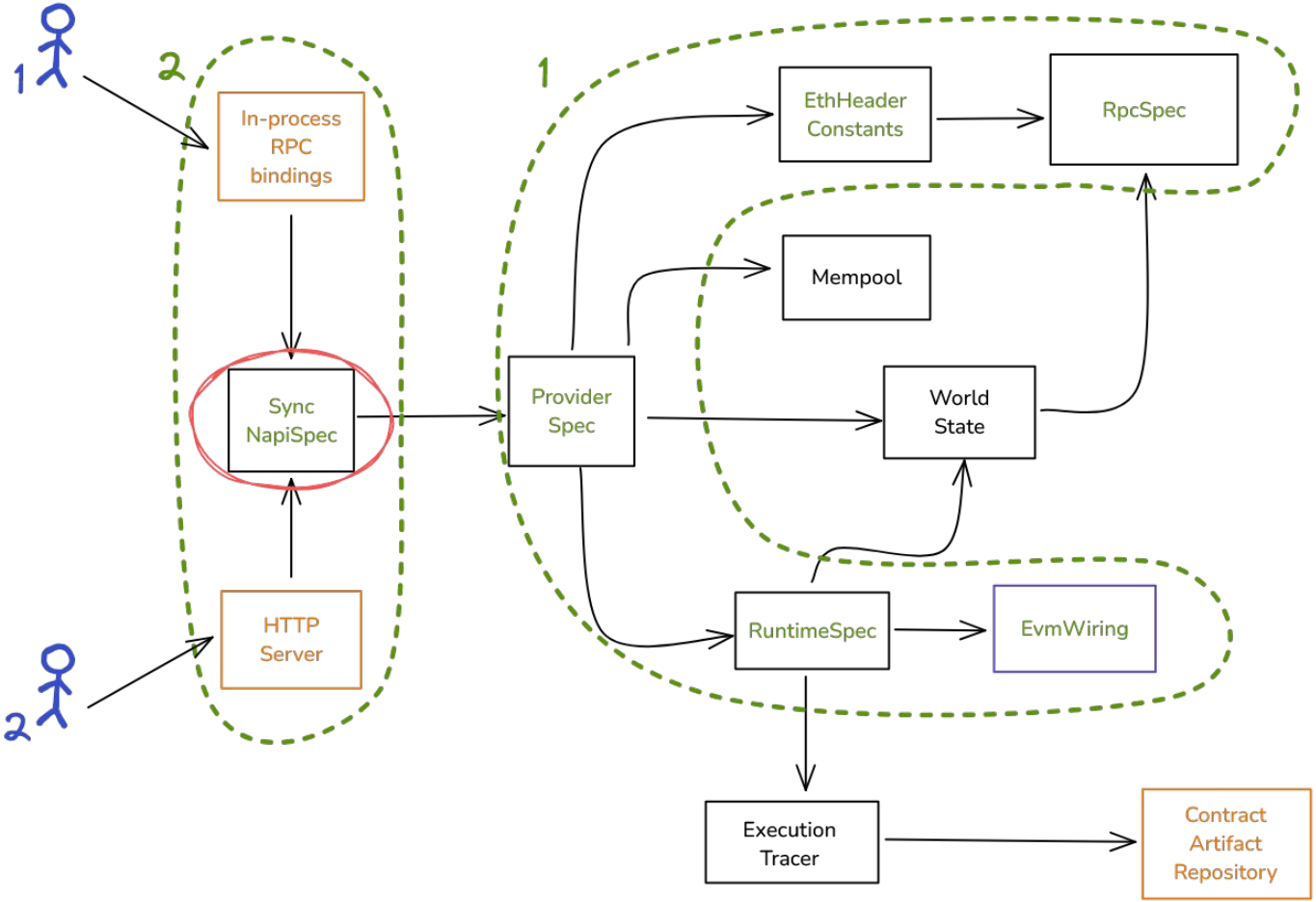

1
2
In-process RPC bindings
RPC Interface
HTTP Server
2
1
Blockchain
RpcSpec
Mempool
Node
World State
RuntimeSpec
EvmWiring
Execution Tracer
Contract Artifact Repository

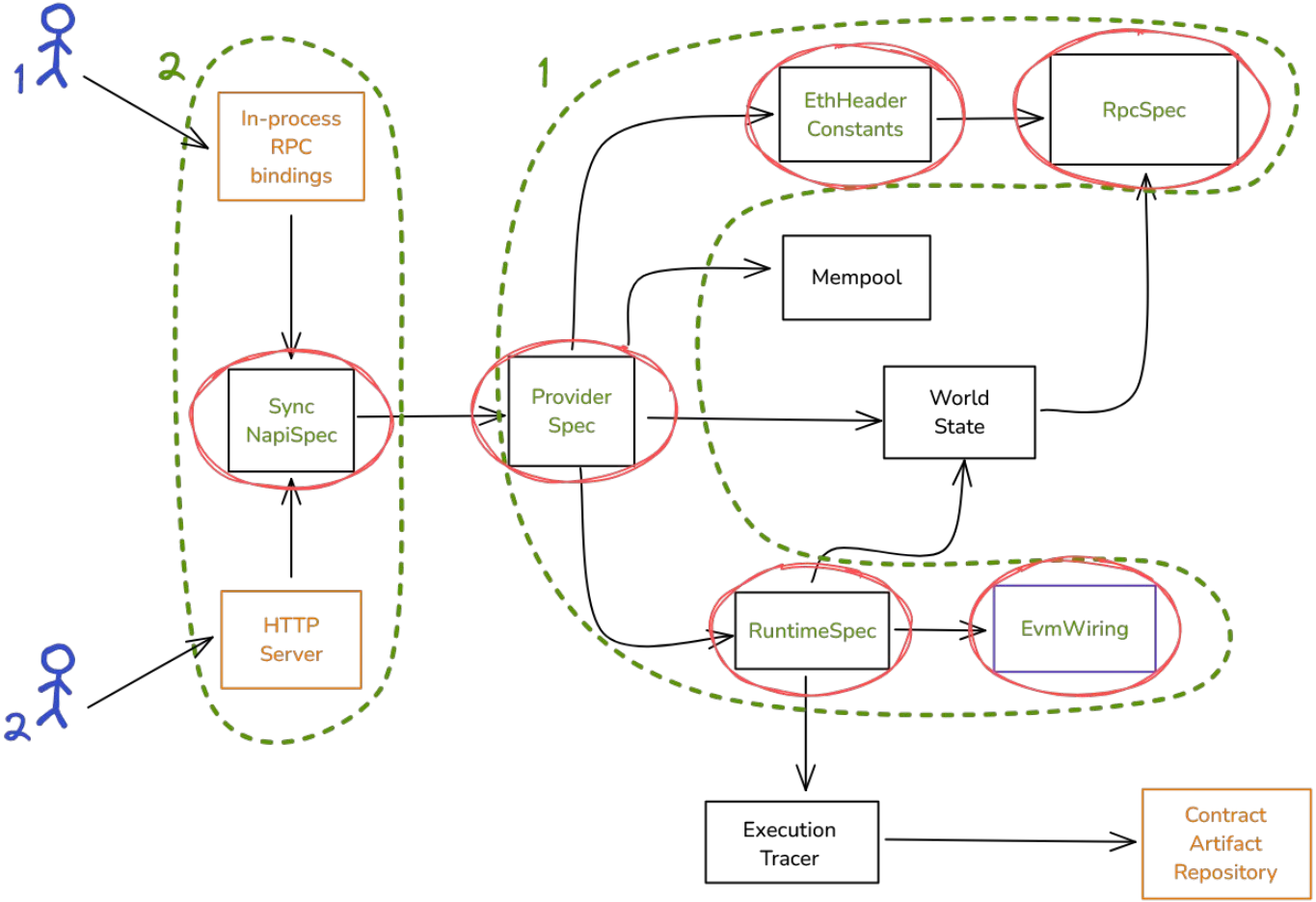

1
2
In-process RPC bindings
RPC Interface
HTTP Server
2
1
EthHeader Constants
RpcSpec
Mempool
Node
World State
RuntimeSpec
EvmWiring
Execution Tracer
Contract Artifact Repository


1
2
In-process RPC bindings
RPC Interface
HTTP Server
2
1
EthHeader Constants
RpcSpec
Mempool
Node
World State
RuntimeSpec
EvmWiring
Execution Tracer
Contract Artifact Repository


1
2
In-process RPC bindings
RPC Interface
HTTP Server
2
1
EthHeader Constants
RpcSpec
Mempool
Provider Spec
World State
RuntimeSpec
EvmWiring
Execution Tracer
Contract Artifact Repository

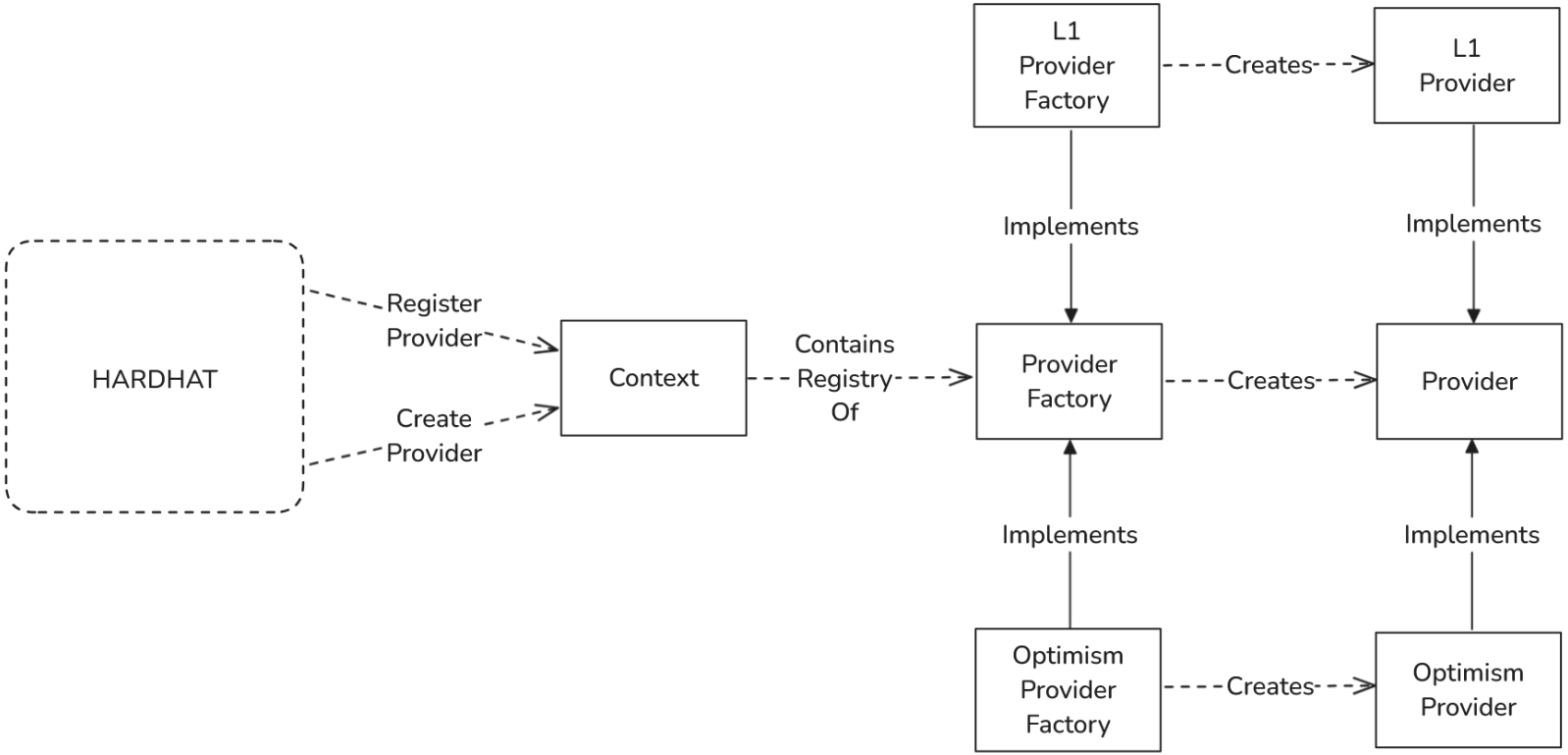

1
2
In-process RPC bindings
RPC Interface
HTTP Server
2
1
EthHeader Constants
RpcSpec
Mempool
Provider Spec
World State
RuntimeSpec
EvmWiring
Execution Tracer
Contract Artifact Repository

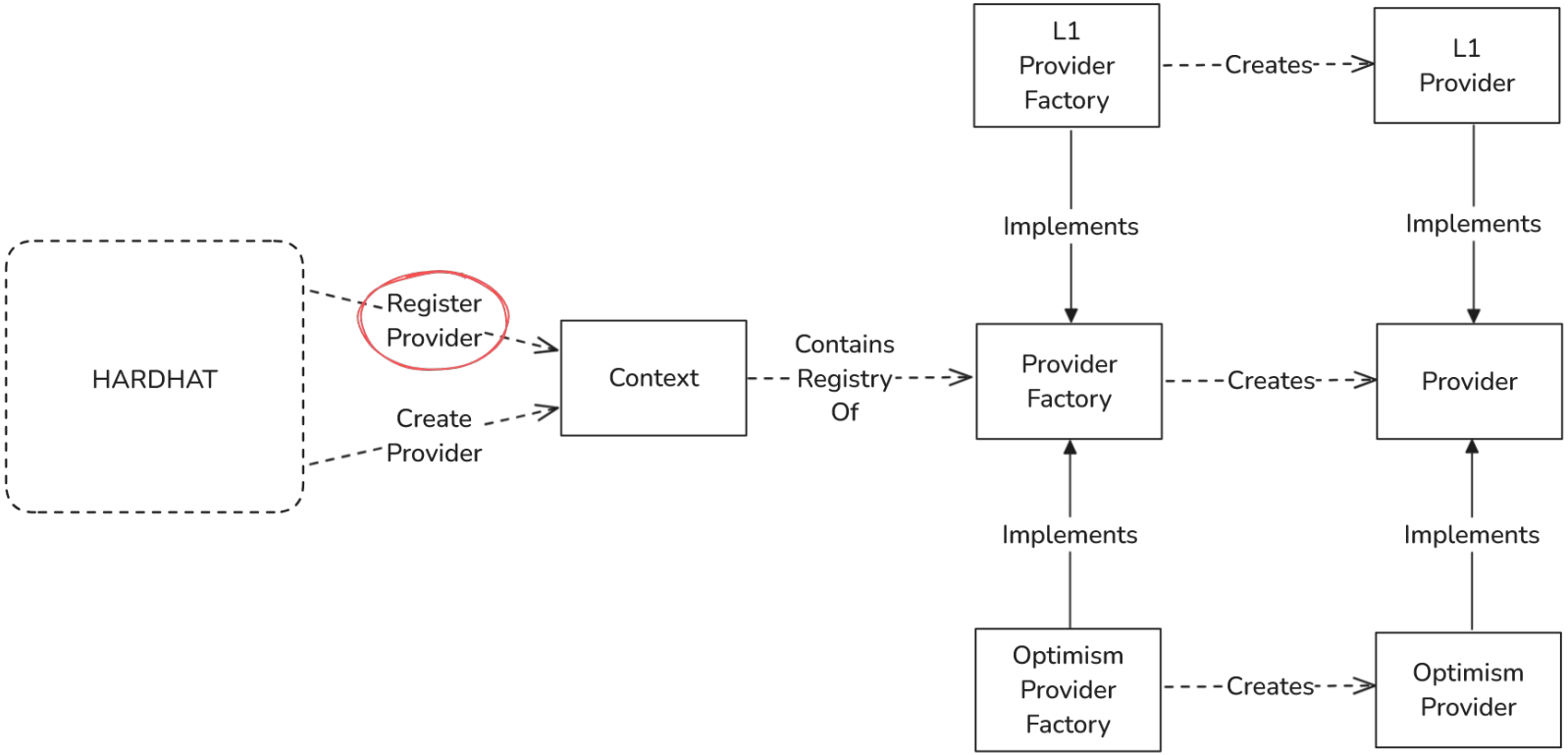

1
2
In-process RPC bindings
Sync NapiSpec
HTTP Server
2
1
EthHeader Constants
RpcSpec
Mempool
Provider Spec
World State
RuntimeSpec
EvmWiring
Execution Tracer
Contract Artifact Repository

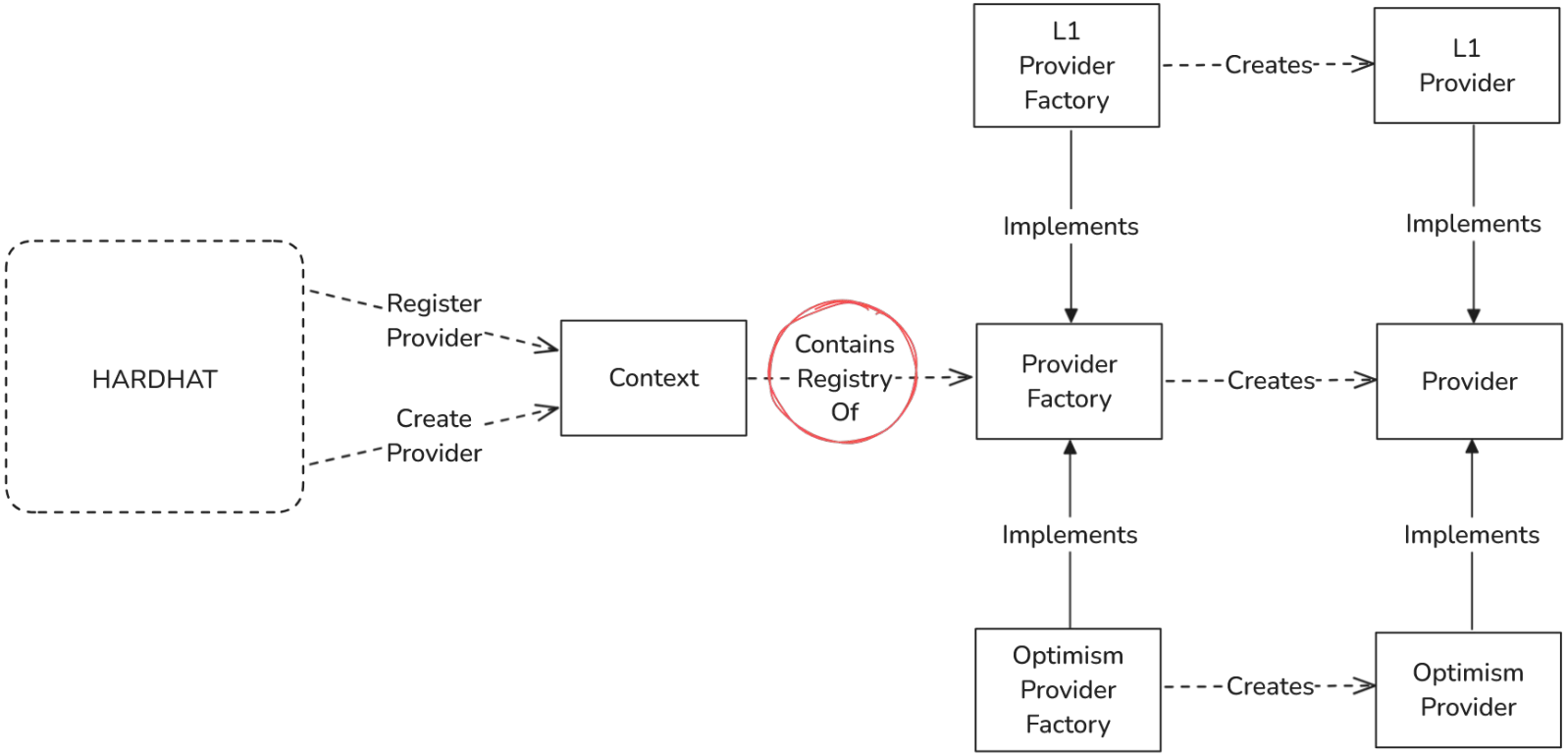

1
2
In-process RPC bindings
Sync NapiSpec
HTTP Server
2
1
EthHeader Constants
RpcSpec
Mempool
Provider Spec
World State
RuntimeSpec
EvmWiring
Execution Tracer
Contract Artifact Repository

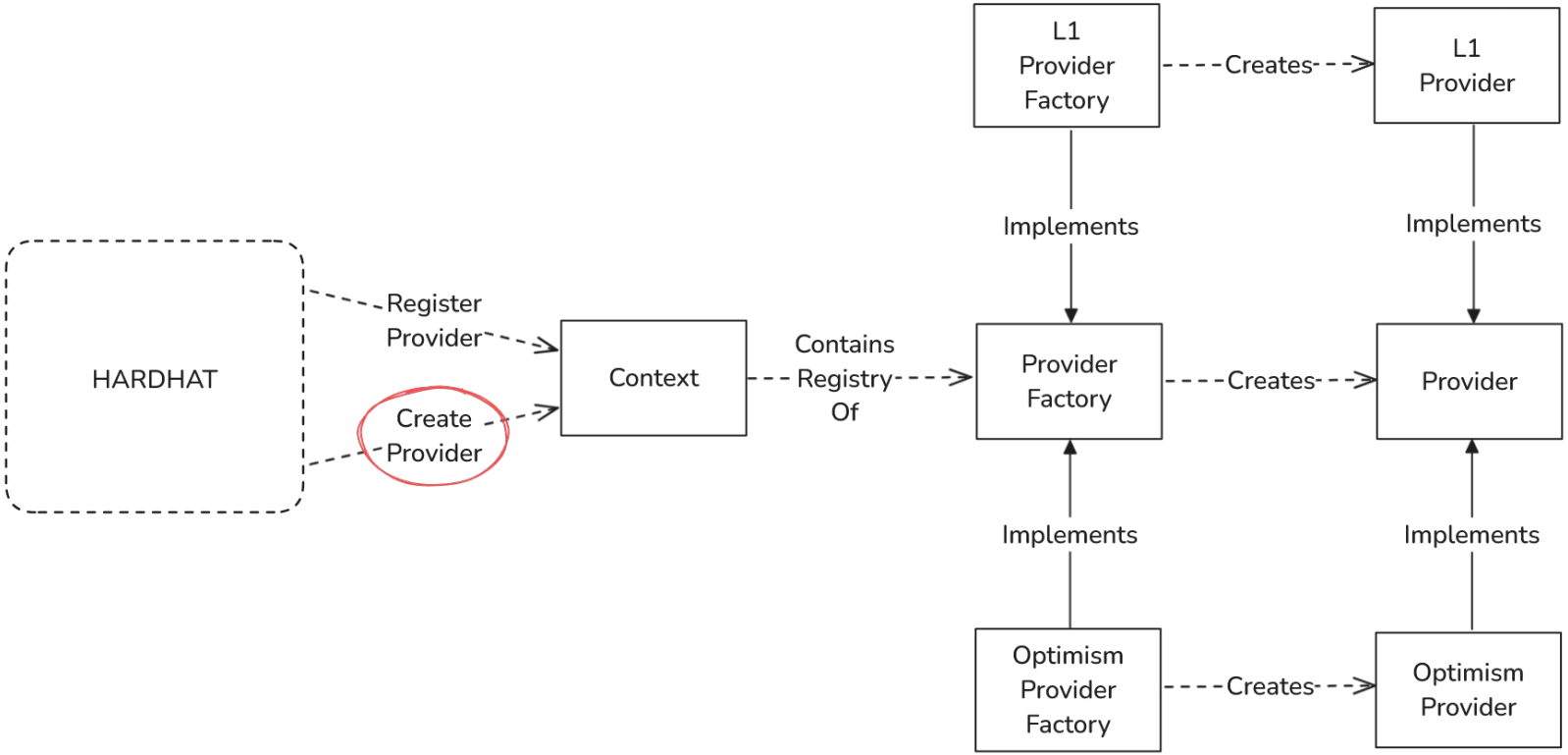
How does EDR simulate L2s?

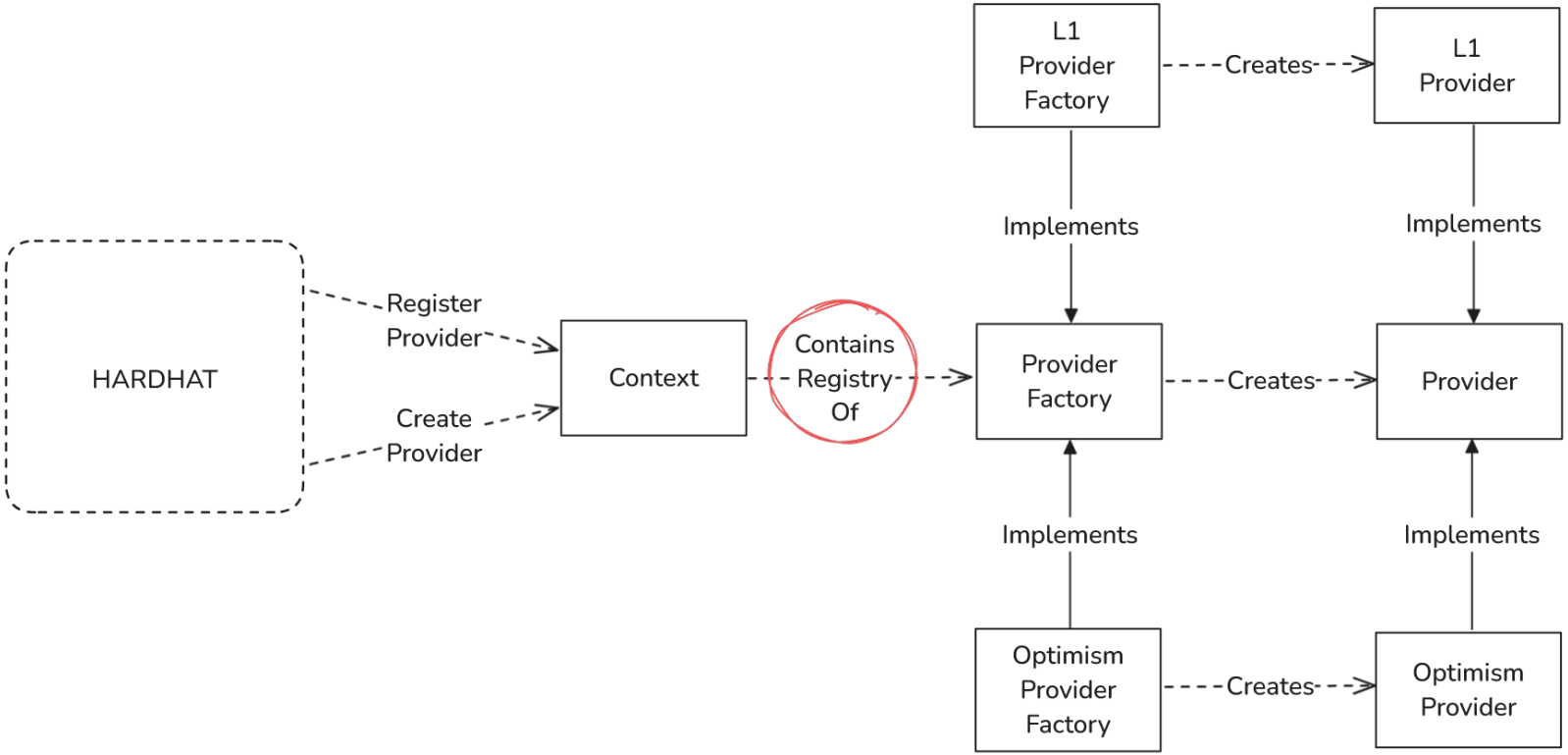
- Extensibility in Typescript
  - Runtime polymorphism
  - Minimise memory usage & load times
  - No centralisation of chain types
- N-API wrapper around dynamic trait objects
  - String identifier for chain types
  - Distribution using NPM packages

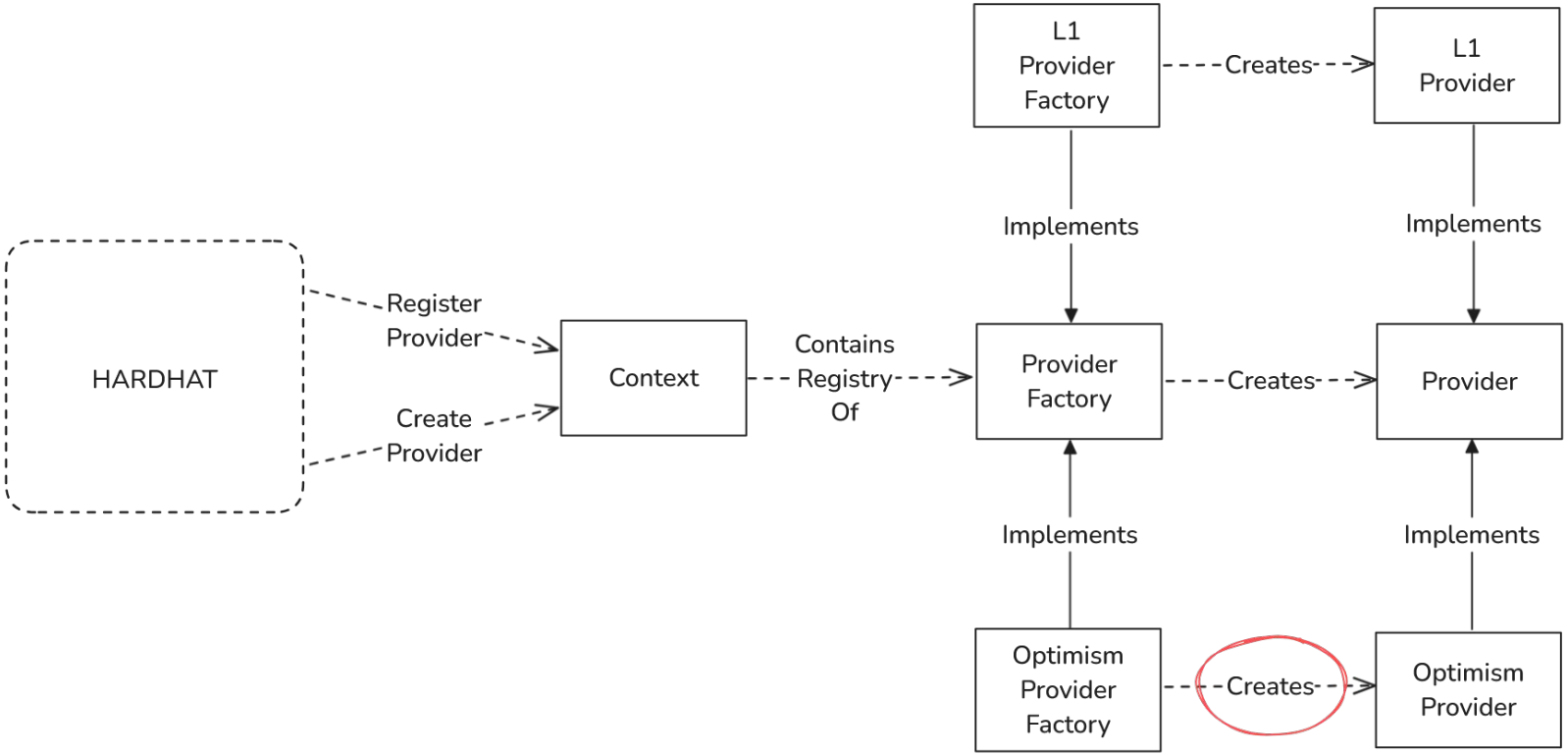

HARDHAT
Register Provider
Create Provider
Context
Contains Registry Of
L1 Provider Factory
Creates
L1 Provider
Implements
Implements
Provider Factory
Creates
Provider
Implements
Implements
Optimism Provider Factory
Creates
Optimism Provider


HARDHAT
Register Provider
Create Provider
Context
Contains Registry Of
L1 Provider Factory
Creates
L1 Provider
Implements
Implements
Provider Factory
Creates
Provider
Implements
Implements
Optimism Provider Factory
Creates
Optimism Provider


L1
Provider
Factory
Creates
L1
Provider
Implements
Implements
HARDHAT
Register
Provider
Create
Provider
Context
Contains
Registry
Of
Provider
Factory
Creates
Provider
Implements
Implements
Optimism
Provider
Factory
Creates
Optimism
Provider


HARDHAT
Register Provider
Create Provider
Context
Contains Registry Of
L1 Provider Factory
Creates
L1 Provider
Implements
Implements
Provider Factory
Creates
Provider
Implements
Implements
Optimism Provider Factory
Creates
Optimism Provider


L1
Provider
Factory
Creates
L1
Provider
Implements
Implements
HARDHAT
Register
Provider
Create
Provider
Context
Contains
Registry
Of
Provider
Factory
Creates
Provider
Implements
Implements
Optimism
Provider
Factory
Creates
Optimism
Provider


L1
Provider
Factory
Creates
L1
Provider
Implements
Implements
HARDHAT
Register
Provider
Create
Provider
Context
Contains
Registry
Of
Provider
Factory
Creates
Provider
Implements
Implements
Optimism
Provider
Factory
Creates
Optimism
Provider

05 | L2s in Hardhat 3

File Edit Selection View Go Run ··· hh3-tests [Dev Container: EDR @ desktop-linux]

hardhat.config.ts | send-op-tx.ts | zsh hh3-tests

hardhat.config.ts > ...

```ts
import { HardhatUserConfig } from "@ignored/hardhat-vnext/config";

import HardhatViem from "@ignored/hardhat-vnext-viem";

const config: HardhatUserConfig = {
  plugins: [HardhatViem],
  networks: {
    edrL1Sepolia: {
      type: "edr",
      chainType: "l1",
      forkConfig: {
        jsonRpcUrl: "https://sepolia.drpc.org",
      },
    },
    edrOpSepolia: {
      type: "edr",
      chainType: "optimism",
      forkConfig: {
        jsonRpcUrl: "https://sepolia.optimism.io",
      },
    },
  },
};

export default config;

```

Dev Container: EDR @ desktop-linux ⊗ 0 ⚠ 0 0 Live Share 0.53% 0.00 GHz 96% ··· 2.32/9.79 GB Ln 26, Col 1 Spaces: 2 UTF-8 LF {} TypeScript Prettier

Learn more!

Nomic talks @ Devcon

Work with us