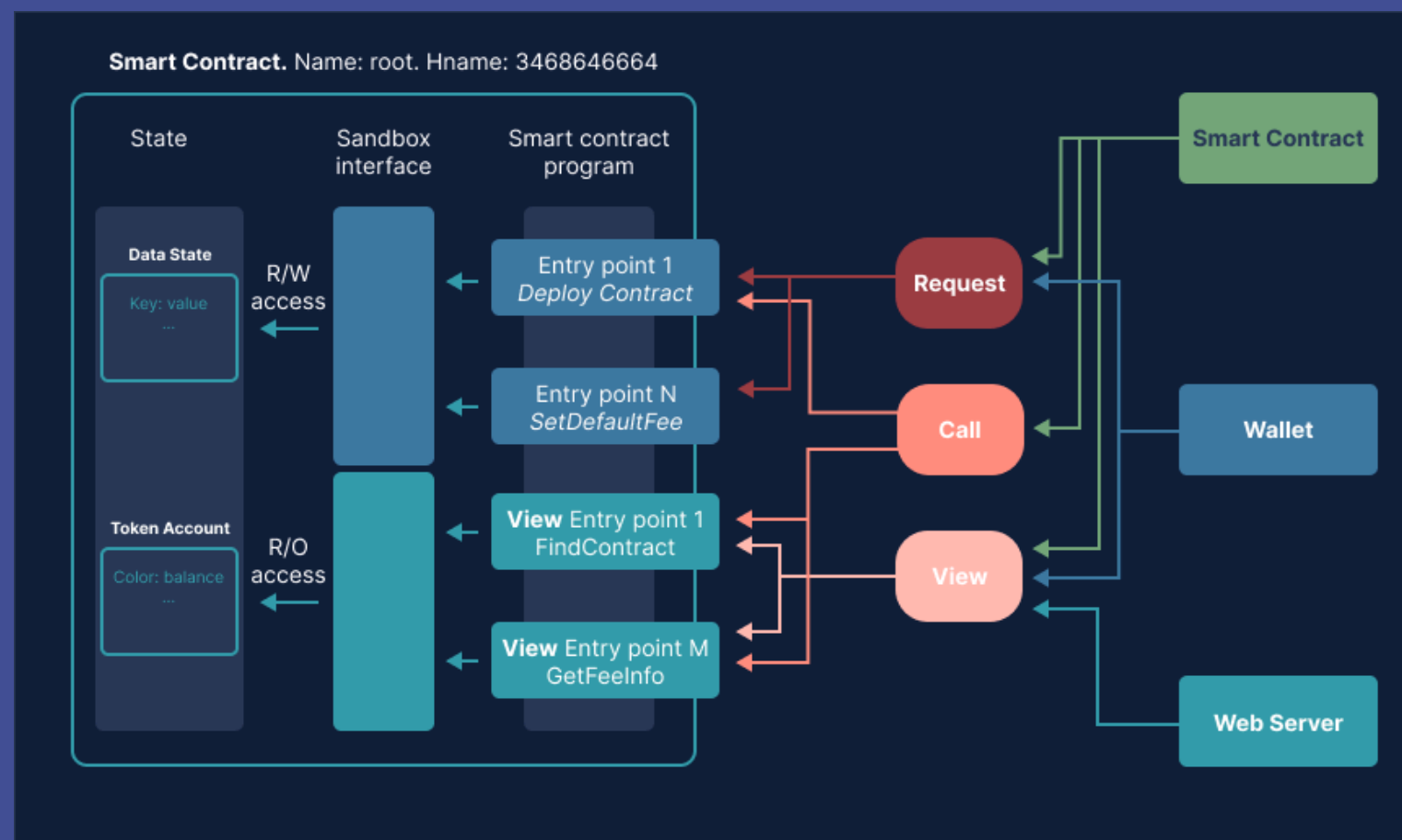


Smart Contracts: Automation on the Blockchain

What are Smart Contracts?

- Self-executing contracts: Contract terms are written directly into code.
- Immutable and transparent: Once deployed, the contract cannot be altered, ensuring trust.
- Automation: Automatically executes actions when pre-defined conditions are met.



Basics of Solidity Programming

- Solidity: Primary programming language for writing smart contracts on Ethereum.
- Syntax similar to JavaScript.
- Supports inheritance and libraries.

Core Concepts:

- Functions: Define the logic of the contract.
- State Variables: Store contract data on the blockchain.
- Events: Log important contract interactions.
- Deployment: Contracts are compiled and deployed on the blockchain.

Tools for Smart Contract Development

Use Cases of Smart Contracts

- Business Process Automation: Streamline operations such as payments, supply chains, and approvals.
- Legal Agreements: Automate contracts like leases, insurance claims, and NDAs.
- Decentralized Applications (DApps): Create applications that run on decentralized networks, ensuring transparency and security.
- Tokenization and NFTs: Managing digital assets and non-fungible tokens securely on the blockchain.

- Remix IDE: A web-based integrated development environment to write, compile, and deploy smart contracts.
- Ethereum Virtual Machine (EVM): Executes smart contracts and ensures they function as intended.
- Testing Networks (Ropsten, Goerli): Simulate deployment and interaction on test networks before going live on the Ethereum mainnet.

Name: Deep Salunkhe:
Roll No.: 21102A0014

