

Chapter 3

Ethereum Ecosystem and Development Lifecycle

Episode 3.01

Parts of the Ethereum Blockchain

The Ethereum Ecosystem

- dApps
 - Decentralized applications
 - Used to interact with the blockchain
- Blocks
 - Primary component of the blockchain
 - Stores transactions and smart contract code
- Miners
 - Nodes that create blocks and solve consensus

Parts of the Ethereum Blockchain

- Nonce (number used once)
- Uncle (ommer) block
 - Any successfully mined block after the first one
 - Miner gets a discounted reward
- Header
 - Attributes that describe a block
- Body

Episode 3.02

Smart Contracts

Smart Contracts

- Code that runs on the EVM
- Enforce rules when exchanging things of value
 - All transactions must resolve the same way on all nodes
- Programs that execute to carry out transactions
 - Only way to access Ethereum blockchain data
- Turing complete
 - Given enough resources, you can calculate anything

Episode 3.03

Smart Contract Languages

What are Programming Languages?

- Like any other language
 - Vocabulary
 - Syntax
 - Semantics
- A software program is like a recipe for a computer
- Uses a specific language with its own vocabulary
- Set of instructions for a computer to carry out tasks
- Use a language that is familiar, capable of doing what you want it to do, and available

Ethereum Smart Contract Languages

- Mutan
 - Older, deprecated language
- LLL
 - Like Lisp
- Serpent
 - Newer, like Python
- Bamboo
 - Newer, blockchain state transitions
- Viper
 - Newer, simplicity and security
- Solidity
 - Most popular, like JavaScript

Episode 3.04

What are Virtual Machines?

Virtual Machines (VMs)

- Run multiple operating systems in a single computer
- Virtually create a computer within a physical computer
- Create multiple virtual machines within one computer

What Can You Do with Virtual Machines?

- Use computers to do more with less hardware
- Easily switch between operating systems or configurations
- Rollback to undo changes
- Ability to provision up or down depending on needs

What Can You Do with Virtual Machines?

- Can help with energy costs
- Can reduce overall IT acquisition and management costs
- Makes cloud computing possible

Episode 3.05

The Ethereum Virtual Machine

Ethereum Virtual Machine (EVM)

- Ethereum uses its runtime environment as the Ethereum Virtual Machine (EVM)
- Software application
- Creates a closed environment
- Smart contracts run in this environment
- Present on all nodes
- Installed with Ethereum

Ethereum Virtual Machine (EVM)

- All EVM instances play by the same rules
 - Any node that breaks the rules creates a “rogue” blockchain
 - Any EVM deviation is immediately evident
 - The offending EVM’s blockchain is deemed invalid
- The EVM provides smart contract execution
 - Only way to access the Ethereum blockchain

Episode 3.06

Fueling Your Code with Gas

What is Gas?

- Cryptocurrency cost of accessing the blockchain
- Incentivize storing local blockchain copies and mining
- Transaction fee
- Set max amount of gas
 - Incentivizes developers to create efficient smart contracts
- Reduces malicious code and DoS attacks
- Increases cost to access the blockchain

Gas Prices

- Gas used
- Gas price
- Formula
 - $\text{Gas used} \times \text{gas price} = \text{total gas cost}$
- Wei units
 - 1 ETH = 1e18 Wei units (1,000,000,000,000,000,000)

Episode 3.07

Tools for Ethereum SDLC

The Software Development Life Cycle (SDLC)

- SDLC steps
 1. Planning
 2. Coding/development
 3. Testing
 4. Deploying
- Ethereum SDLC tools
 - Ethereum blockchain client
 - Development/testing blockchain
 - Compiler/testing framework
 - Source code editor/IDE

Episode 3.08

Blockchain Client

Blockchain Client

- Software that makes a device a node on the Ethereum blockchain
- Supports Ethereum standards
- Runs the EVM on a node
 - Full nodes store the whole blockchain
 - Light nodes store only a portion
- Different clients use different languages

Example Blockchain Clients

Name	Language	Where to Get It
Cpp-ethereum	C++	https://github.com/ethereum/aleth
Ethereumjs-lib	JavaScript	https://github.com/ethereumjs/ethereumjs-lib
Geth (go-Ethereum)	Go	https://geth.ethereum.org/
Parity	Rust	https://www.parity.io/ethereum/
Pyethapp	Python	https://github.com/ethereum/pyethapp

Episode 3.09

Writing and Testing Your Code, Part 1

Local Simulated Blockchains

- Local blockchain
- Used for development and testing
- Never deploy to mainnet without testing
- Same as traditional development
 - Write and test code in safe, simulated environment

Local Test Blockchains

- You have control
- You can delete the whole blockchain and start over
- You don't affect anyone else

Public Test Blockchains

- Ex: Ropsten, Rinkeby, Kovan
- Shared with other users
- Don't have to pay real money

Example Development/Testing Blockchains

Name	Language	Where to Get It
Ganache	Most popular tool with developers to easily create a private network	https://www.trufflesuite.com/ganache
Truffle	Suite of development tools that includes its own private network	https://www.trufflesuite.com/
Cliquebait	Uses docker instances to simulate a real blockchain network	https://github.com/f-o-a-m/cliquebait
Local Ethereum Network	Easy to use scripts to setup private blockchain networks	https://github.com/ConsenSys/local-ethereum-network

Episode 3.09

Writing and Testing Your Code, Part 2

Source Code Editor/IDE

- Built-in compiler
- Syntax checking and assistance
- Organization and collaboration
- Integrated development environment (IDE)
 - One workspace with several views and tools

Integrated Development Environments (IDEs)

Name	Language	Where to Get It
Atom	Popular, has Solidity plugins	https://atom.io/
Visual Studio Code	Microsoft, has Solidity plugins	https://marketplace.visualstudio.com/items?itemName=JuanBlanco.solidity
Vim Solidity	Solidity plugins for Vim (vi-like editor)	https://github.com/tomlion/vim-solidity
Remix	Web-based, popular with new Solidity developers	https://remix.ethereum.org/
EthFiddle	Web-based, simple	https://ethfiddle.com/
Superblocks Lab	Web-based, many blockchain integration features	https://lab.superblocks.com/

Frameworks

- After writing source code you need to
 - Compile source code into bytecode
 - Deploy to a development or testing blockchain
 - Run your code to test it
- Manually completing deployment and testing tasks is tedious
- Frameworks help to simplify smart contract management

Development and Testing Frameworks

Name	Language	Where to Get It
Truffle	Manage smart contract development, testing, deployment	https://www.trufflesuite.com/
Solidity compiler (solc)	Solidity software, includes CLI compiler that can be called from IDEs	https://github.com/ethereum/solidity
Solidity compile (solcjs)	Solidity compiler written in JavaScript	https://github.com/ethereum/solc-js
Remix	Web-based, suite of development tools, includes Solidity compiler	https://remix.ethereum.org/
Populus	Web-based IDE for smart contract development	https://github.com/ethereum/populus
Embark	Framework for developing dApps for multiple blockchains	https://github.com/embark-framework/embark