Decentralized Applications (Dapps): Key Takeaways

Decentralized Applications - Key Takeaways

Below you will find a number of key points from this course. Defined terms are underlined.

Week One: Decentralized Applications (Dapps)

A Dapp, or decentralized application, solves a problem that requires blockchain services and blockchain infrastructure for realizing its purpose.

Blockchain server represents the infrastructure and the functionality the blockchain provides.

The design of a Dapp has a ____ front-end, a blockchain back-end and the code connecting the two.

API or Application Programming Interface is a convenient and standard way to expose a set of functions related to a specific data set and services.

An API publishes a set of functions or methods that can be used programmatically to invoke operations, access data and store data.

Two well-known examples of API are:

- Twitter API to access tweets that can be filtered by query terms

- Google Maps API that allows for applications to “embed” the map features such as geolocation in their own applications, leveraging and reusing the power of google map API.

There are two major categories of APIs:

- The first one is for management APIs that includes admin, debug, miner, personal, and txpool. They support methods for management of the geth node.

- The second one is the Web3 APIs: web3, eth and net. They support methods for development of Dapps.

Week Two: Truffle Development Environment

Applying the general design process requires five steps:

- Step 1: Design the Ballot.sol;

- Step 2: Illustrate modifiers with just one modifier “onlyOwner” referring to the qualified person as the chairperson. Recall that in the design of a smart contract you can use modifiers to represent global rules.

- Step 3: Add a tester code for the required problem specification and run the tests to make sure they all pass.

- Step 4: Add the user interface component and

- Step 5: Test the complete application by interacting with the interface.

Basic Truffles Commands:

- Truffle Init: Initializing a template or base directory structure for a Dapp

- Truffle compile: Smart contract compilation

- Truffle develop: Personal blockchain for testing with a console

- Truffle migrate: Migration scripts for deploying smart contracts

- Truffle test: Test environment for testing the deployed contract

Metamask will link to the local blockchain created by Truffle to manage the accounts, acting as a bridge between the application front-end and the blockchain node that hosts the accounts.

Node.js which will serve as your web server for the Dapp front-end

Smart contracts are like your hardware chip. Once deployed, they are final and cannot be updated unless special provisions or escape hatches are built-in.

Positive tests - making sure, for a given valid input, it performs as expected. We will test the complete ballot cycle of deploy-register-vote-winningProposal

Negative tests - making sure it handles invalid inputs or situation appropriately. We will code only 2 of the many negative tests possible.

test.js has 4 positive tests and 2 negative tests.

Asserts in the positive tests have three parameters: expected value, actual value, a string representing the test being performed.

Contracts has the solidity contracts

Migrations has the migration scripts

Test has the test scripts

Build has the json artifacts generated by the compile process

Src has the web assets such as js, css, index.html

Node_modules has the node.js modules

The **json** files and **js** files are configurations files.

Week Three: Improving the Smart Contract Design

Memory is transient memory in RAM

Storage refers to the persistent store in a permanent storage device like your hard drive.

Solidity provides a function called **"selfdestruct"** to delete or kill a smart contract.

Libraries are special smart contracts with no Ether balance, no payable functions, and no state to be stored on the blockchain.

A smart contracts ownership may change or deleted

Coin sc has two data items.

- The first one is the address of the minter or owner;
- The second is the mapping between the account address and the COIN balance

Coin has a mint function that only the minter can execute to mint new coins for a given address. You have to be in Account1 in Metamask or minter's account to mint.

Coin has a transfer function that can transfer Coins from one account to another. You have to be in Sender's account in Metamask to be able to transfer.

It is possible, a complex application could be composed of many smart contracts; one instantiating the other or inheriting others

A transfer function may be used to implement a transfer in ownership of the Smart contract.

Oraclize is described as a data carrier between the web resources (APIs, and URLs) and a smart contract.

usingOraclize is a smart contract that provides minimally a query function to access external sources.

Week Four: Application Models and Standards

Initial Coin Offering (ICO) uses a blockchain to record the distribution of the Coin, receive funds, specify rules, and to enforce any preconditions and policies.

Token is similar to a Dapp Coin, but its offering is typically associated with an asset or utility.

Decentralized Autonomous Organization (DAO) is an investment instrument deployed as a smart contract on Ethereum blockchain whose main goal is to showcase an autonomous organization without traditional corporate governance structure for decentralized, anonymous fundraising and automatic investing.

The **Decentralized Marketplace** facilitates meetings of sellers and buyers.

Fintech, short for Financial Technology, has great potential for innovative Dapps whose domain ranges from decentralized investment instruments to micropayment channels.

EIP or Ethereum Improvement Proposal is a means to manage the protocol specification, improvements, updates, client APIs and contract standards.

EIP handles issues in four different categories, including:

- Core or core ethereum protocol
- Network or network level improvement
- Interface or Interfaces such as ABI, RPC and
- ERC or Application level convention and standards

Ethereum Request for Comments (ERC) is a solution draft proposed by the ERC document and discussed on the github, gitter and sub-reddits community.

Download and print this document:

Decentralized-Applications-Key-Takeaways-Page.pdf
PDF File