Decentralized Applications (Dapps)

Provides blockchain features and services to the outside world for review, interactions and enjoyment.

Gives access to the blockcahin for people and applications and systems, not necessarily known to each other to transact peer-to-peer.

It is an end-to-end application development process

Decentralized Apps Stack

- Verticals: End User Applications
- Application Framework: Smart Contracts
- Ethereum Blockchain and Ethereum Virtual Machine (EVM)
- Peer-to-Peer Network and Operating Systems
- Hardware

snippet.shell

```
sudo apt-get install software-properties-common
sudo add-apt-repository -y ppa:ethereum/ethereum
sudo apt-get update
sudo apt-get install ethereum
```

Ethereum blockchain protocol services: Blockchain Server

snippet.shell

```
geth --datadir ./ account new
geth --datadir ./ init customgenesis.json
geth --datadir ./ --maxpeers 95 --networkid 15 --port 303xx console
```

Dapp Defined

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

Here blockchain server is the Ethereum 'node' and uderlying infrastructure, and client or the front-end is a web client with embedded web3.js script, communicating using JSON over RPC pipeline.

Truffle

truffle.js - Contains truffle deployment configurations information

Add a filel to the migrations directory to deploy our smart contract:

2*deploy*contracts.js

snippet.javascript

```
var Ballot = artifacts.require('Ballot');
module.exports = function(deployer){
  deployer.deploy(Ballot);
}
```

You can simply copy the file 2*deply*contracts.js from Resources for the course, into the migrations folder/directory

What can you do with truffle console?

This console provides a command line interface to the accounts created on the the test chain and also to the management APIs we discussed earlier in Module 1.

Testing of a smart contract can be done by:

Positive tests - Given valid input, a smart contract performs as expected

Negative tests - For negative inputs, a smart contract captures the errors

Test-Driven Development

truffle compile truffle migrate –reset

Web Interface & Testing

In the truffle IDE default port for the test chain deployed by “Truffle develop” command is 9545.

관련 문서

- BloakChain

From:
http://moro.kr/ - **Various Ways**

Permanent link:
http://moro.kr/open/dapps

Last update: **2020/06/02 09:25**