upGrad

#LifeKoKaroLift

DeFi Smart Contract

upGrad

Course: PGD Software Development (Blockchain)

Lecture On: Defi Smart Contract

Instructor: Jitender Bhutani

upGrad

Topics covered in the previous class...

1. What is DeFi?
2. Centralized vs Decentralized
3. Discuss live use cases
    a. Uniswap
    b. Aave
    c. MakerDAO
    d. Compound
    e. Binance
4. DeFi app architecture

upGrad

Overview

- Session 1 : Understand the Architecture of DeFi Application
- **Session 2 : Develop DeFi App Smart Contract**
- Session 3 : Develop DeFi App Backend
- Session 4 : Further scope
  - Frontend
  - Upgradable smart contract
  - Administration
  - Scalability

Today's Agenda

- Create and Build a Defi App from scratch
  - Start Private Blockchain Network
  - Write Smart Contract
    i. Define Functions
    ii. Define Request Factory
    iii. Security Measures
  - Write Test Cases
  - Deploy Smart contract
    i. Private network
    ii. Ropsten
  - Discuss Design Patterns

upGrad

Poll 1 (15 seconds)

Which of the below is the name of ethereum test network?

A. Ropsten

B. Rinkeby

C. Kovan

D. All of the above

Poll 1 (15 seconds)

Which of the below is the name of ethereum test network?

A. Ropsten

B. Rinkeby

C. Kovan

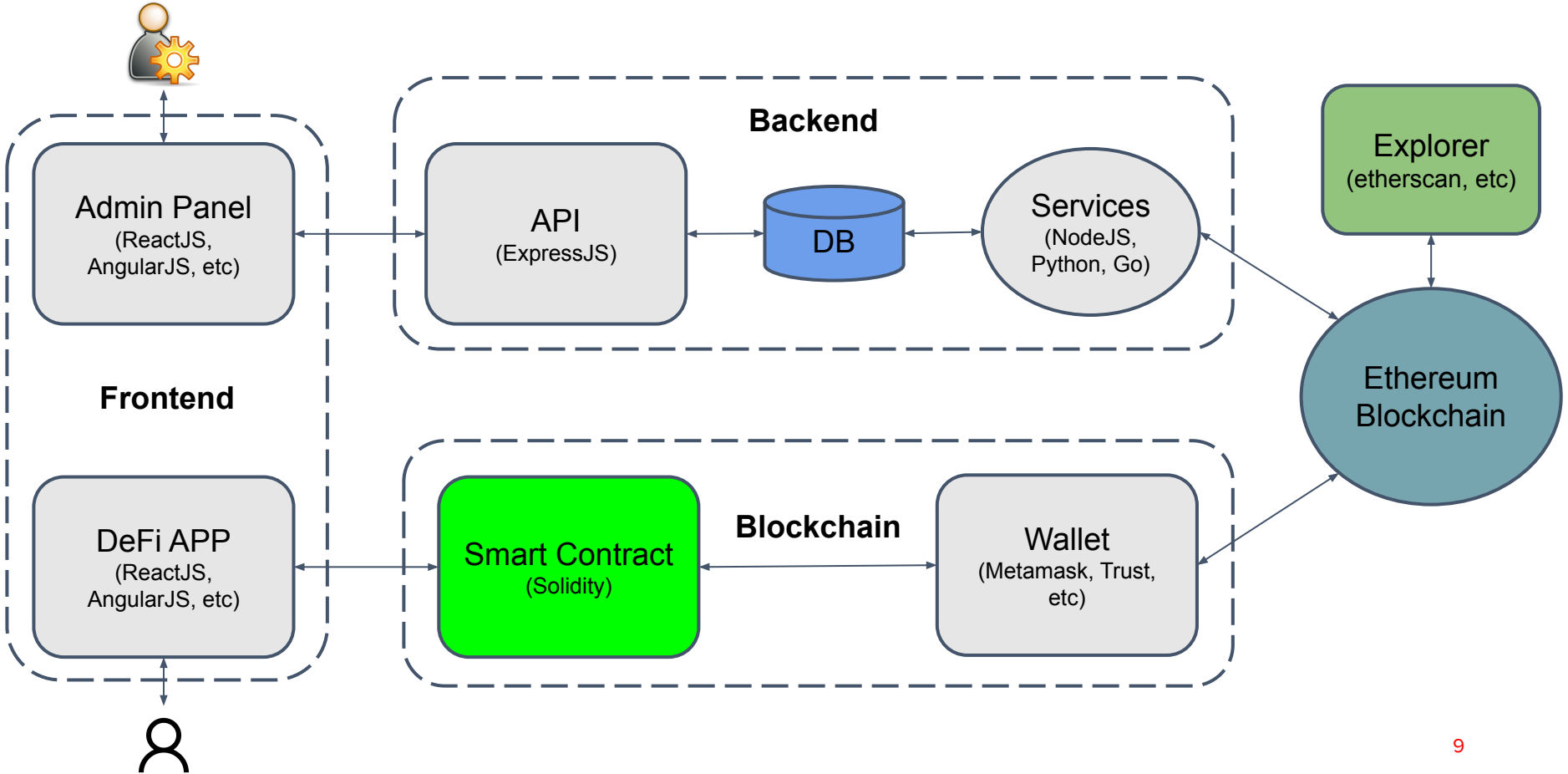
D. All of the above

upGrad

SMART CONTRACT DEVELOPMENT

DeFi ARCHITECTURE

upGrad



APPLICATION LAYERS

upGrad

- **Smart Contract**

  Holds the business logic of DeFi

  - ERC20 token
  - Ask Token (With Collateral)
  - Lend Token
  - Payback
  - Collect Collateral
  - Cancel Request

upGrad

DESIGN PATTERNS

- **Contract Self Destruction**
  - To terminate a contract
  - To remove it forever from blockchain
  - Once destroyed
    - Not possible to **invoke** functions on the smart contract
    - No transactions will be logged in the ledger
  - Syntax
    - selfdestruct(address)
  - Sends all of the contract's current **balance** to address
  - Reduce your gas costs via **negative** gas because the operation **frees** up space on the blockchain by clearing all of the contract's data

- **Factory Contract**
  - For storing **child** contracts addresses so that they can be extracted whenever necessary
  - Can be used to create and manage assets attributes like (eg. who is the owner)
  - Example :
    - address newCarAsset = new CarAsset(...)
  - Can you store them in ur **app database** instead of Blockchain?
    - Yes you can store, however by storing them in the contract, they remain in the blockchain and they are safe there, while data corruption in your database might wipe the asset contracts.

- **Name Registry**
  - Useful when you have to interact with a lot of contracts on blockchain and their **address** can **change** over time
  - It is done by storing a **mapping** contract **name** => contract **address**
  - And each address can be looked up from within the app
  - Example :
    - getAddress("Contract1")
- **Mapping Iterator**
  - Mapping is required most of the times in contracts, but since mappings in solidity **cannot be iterated** and they can only store values.
  - Mapping Iterator pattern turns out to be very useful
  - Example
    - mapping(string => address) elements;
    - string[] keys;

upGrad

SMART CONTRACT DEMO

upGrad

DOUBT CLEARANCE WINDOW

In this class, you learnt that:

upGrad

1. How to Start Private Blockchain Network
2. Understand DeFI Smart Contract Design Pattern
3. Deploy Smart contract on blockchain
    a. Private network
    b. Ropsten

HOMEWORK

upGrad

1. Deploy contract on ethereum ropsten network
2. Verify and publish the smart contract on etherscan and share the smart contract address
3. Have transaction fee during lending and borrowing in ERC20 and transfer that to smart contract owner address **(Research Based)**

Next Steps

- Create and Build a Defi App from scratch
  - Backend
- Further scope
  - Frontend
  - Administration
  - Scalability

upGrad

#RahoAmbitious

Thank You!