upGrad
#LifeKoKaroLift

DeFi Back End Application

Course: PGD Software Development (Blockchain)

Lecture On: Defi Backend Application

Instructor: Jitender Bhutani

Topics covered in the previous class...

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

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 Backend for DeFi APP
  - Write API for smart contract transactions
  - Manage Key Pairs
  - Create Transaction
  - Sign Transaction
  - Broadcast Transaction

Poll 1 (15 seconds)

Which of the below is true with respect to self destructing a contract?

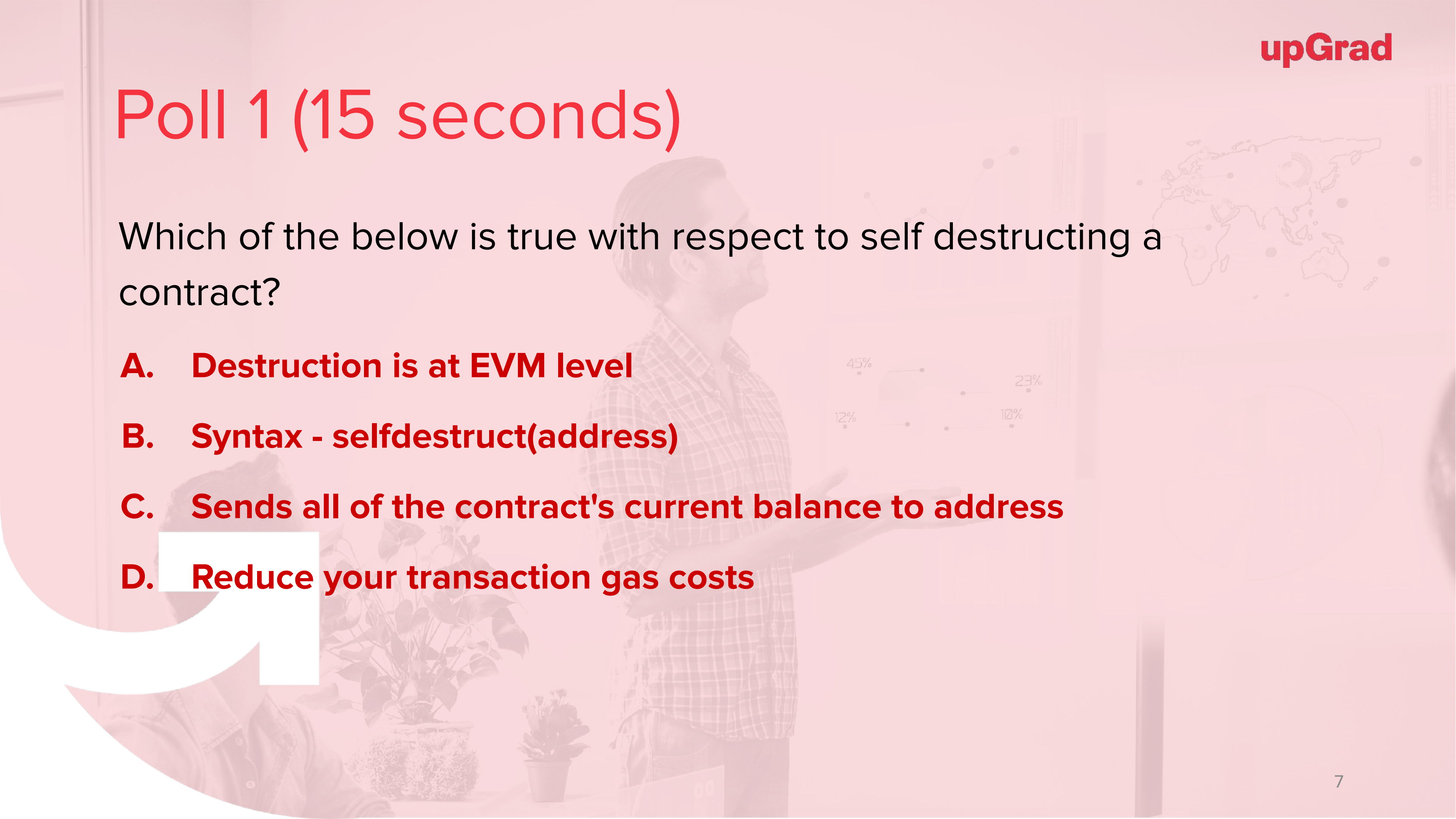
A. Destruction is at EVM level

B. Syntax - selfdestruct(**address**)

C. Sends all of the contract's current balance to **address**

D. Reduce your transaction gas costs

Poll 1 (15 seconds)



Which of the below is true with respect to self destructing a contract?

A. Destruction is at EVM level

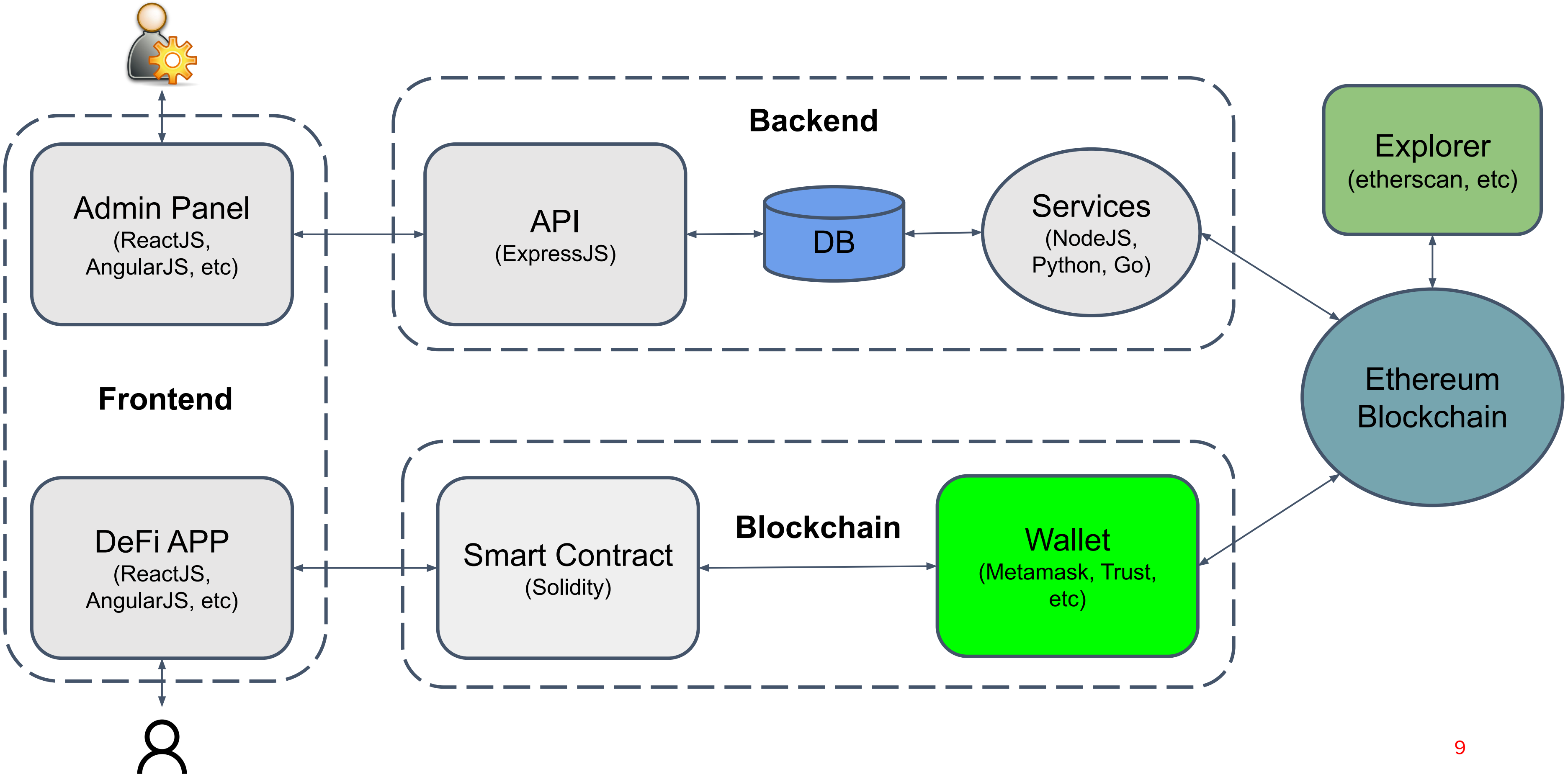
B. Syntax - selfdestruct(address)

C. Sends all of the contract's current balance to address

D. Reduce your transaction gas costs

BACKEND DEVELOPMENT

DeFi ARCHITECTURE



APPLICATION LAYERS

- **Backend**
  - API
    - For both Admin and DeFi APP
    - To view Blockchain Transaction data
    - Analytics
  - Services
    - To store our DeFi App blockchain transaction in local DB

TRANSACTION MANAGEMENT

- **Create Transaction**
  - Call(.call()) smart contract functions
    - For **read** operations (view and pure functions)
      - Example : getRequests().call()
    - Does not require gas(transaction fee)
  - Raw Transaction
    - For **write** operation
    - Require smart contract address and ABI (Application Binary Interface)
    - Input
      - Data(Called function and input parameteres)
      - Gas limit
      - Gas price
      - From address
      - Nonce

TRANSACTION MANAGEMENT

- **Sign Transaction**
  - Once the raw transaction is created, it needs to be signed by “from” address(from raw transaction) **private key**.
- **Broadcast Transaction**
  - After signing transaction is broadcasted to a blockchain node.
  - From there it follows the blockchain transaction life cycle
- **Important**
  - It is possible that your transaction is correct but can still fail during smart contract execution, which leads to paying unnecessary gas.
  - So it is advisable to test smart contract transaction before broadcasting the transaction via **call()** function.
  - Call function will **throw** an **error** if transaction can be failed in smart contract.

BACK END CODING DEMO

DOUBT CLEARANCE WINDOW

In this class, you learnt:

1. Create Backend for DeFi APP
   a. Write API for smart contract transactions
   b. Manage Key Pairs
   c. Create Transaction
   d. Sign Transaction
   e. Broadcast Transaction

HOMEWORK

1. Do batch transaction via api from a single address with proper nonce
2. Write script to capture blockchain transaction specific to your contract

Next Steps

- Further scope
  - Frontend
  - Administration
  - Scalability

upGrad

#RahoAmbitious

Thank You!