# To design and develop end-to-end decentralized applications (Dapps).

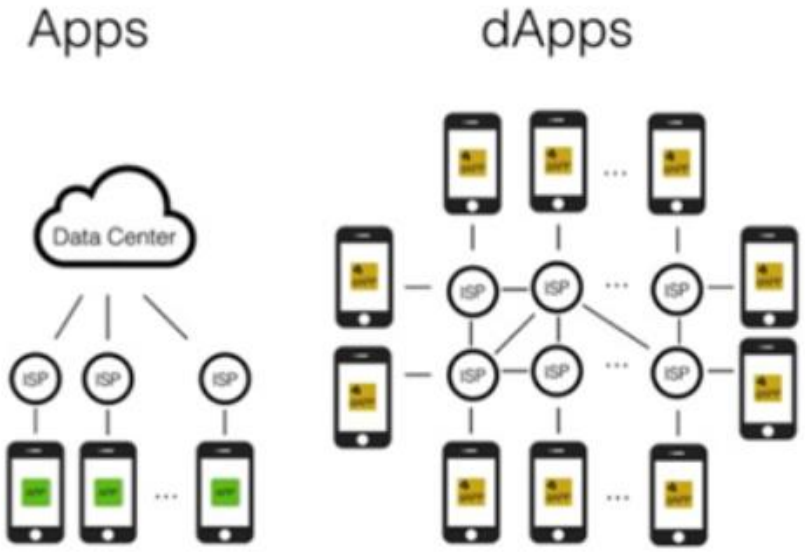
**Practical 8**

# When you’re peering curiously into blockchain….

“I think I want to build a decentralized app, but can’t figure out WHY I actually should or HOW to do it.”

# What is DApp?

- A P2P type of network that is governed by members all around the world.
- Running of the app (and its data) is distributed across lots of people/nodes.
- There is no single (central) point of failure…



# Characteristics of DApp

- **Open Source**: the worldwide availability of the main source code.
- **Decentralization**: the utilization of blockchain technology
- **Incentivization**: the provision of the cryptographic tokens or digital assets to the contributors.
- **Protocol**: the token generation combined with the consensus mechanism.

# Pros and Cons of DApp

- PROS
  - You don't need to trust anyone with your data.
  - No servers needed. With it, no maintenance or rent.
  - 100% uptime, guaranteed.
  - Impossible to backdoor without a million hackers at your disposal due to BFT.
  - Quick and easy to deploy, usable without any frontend.
- CONS
  - Slow, inefficient, riddled with fees for the end user.
  - Immutable, difficult to future-proof.

# Problems with the DApp

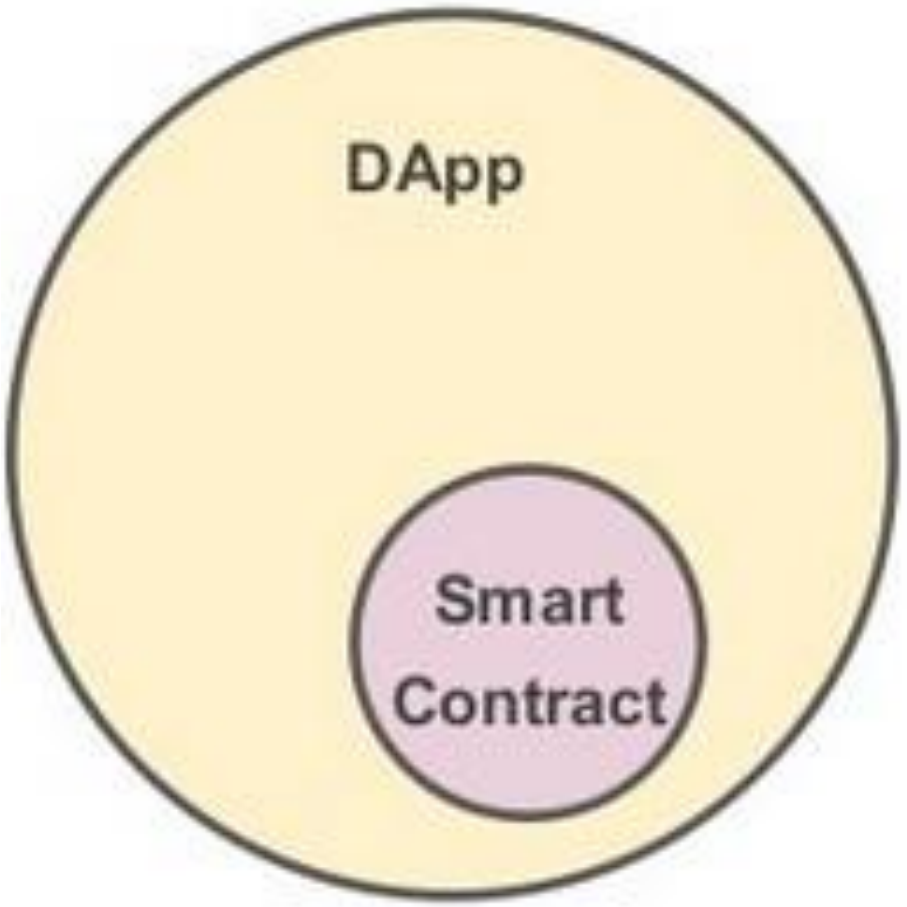
1. **Cost:** Because computation runs on every node, by definition it is at least as costly as the number of nodes in the network (eg 100x amazon AWS).
2. **Time:** Since multiple nodes have to first run the computation (fast) and then come to a consensus about its result (slow), it's much slower than central servers.
3. **Services:** How do you provide services to users?
4. **Privacy:** Right to be forgotten.

*"You can compete on any 2"*


*...unless you're decentralized.*

# DAps vs Smart Contract

- The DApp is the full application including its front-end markup.
- Any non-blockchain code on the front end or a separate server AND the smart contract.
- The smart contract is just the portion of the app that actually works with the blockchain.



# Steps

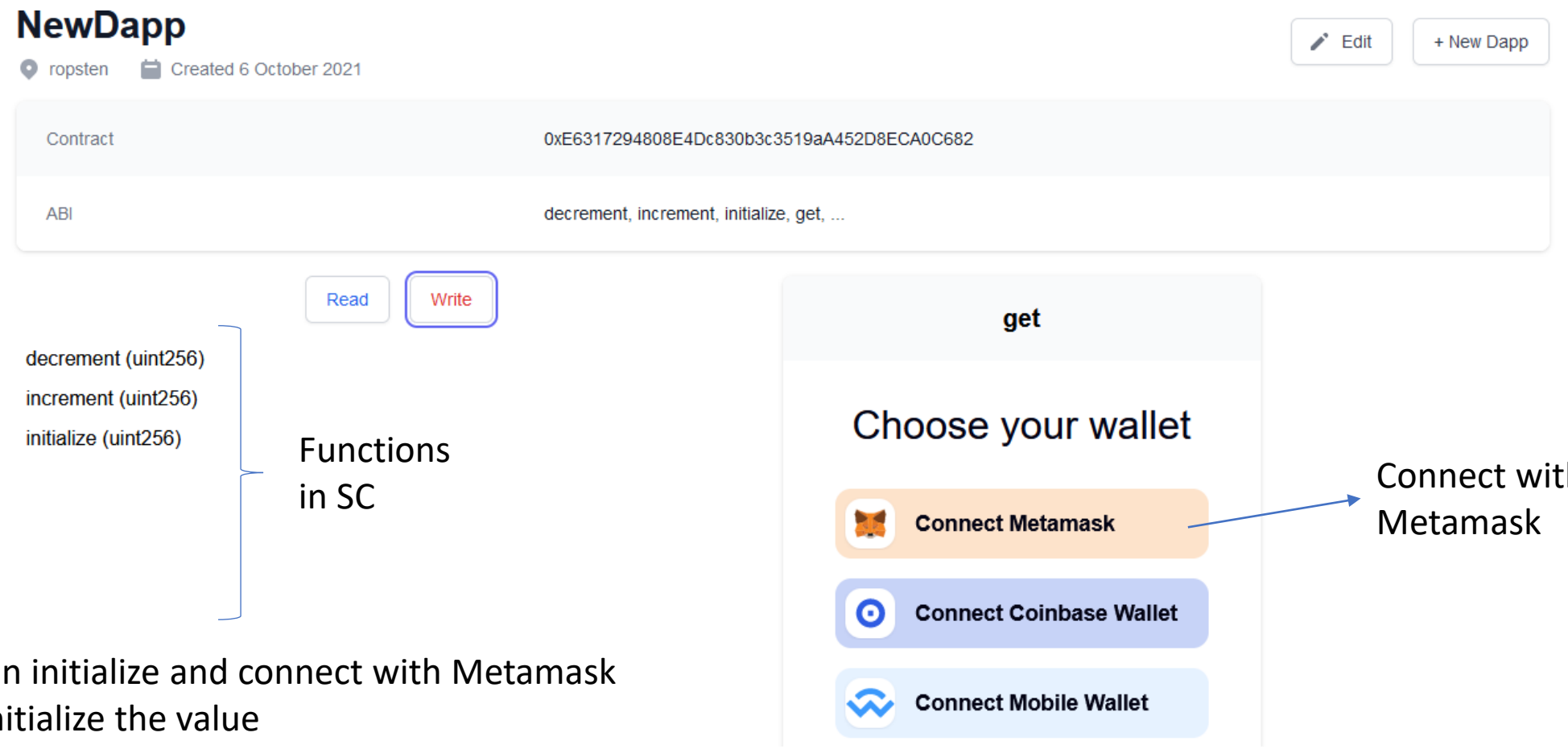
- Create Solidity program in Remix
- Compile it.
- It will generate ABI and bytecode
- Deploy it
- It will generate smart contract address
- Open https://oneclickdapp.com/
- Create DApp by signining in.

# Sample Code

- pragma solidity ^0.6.0;

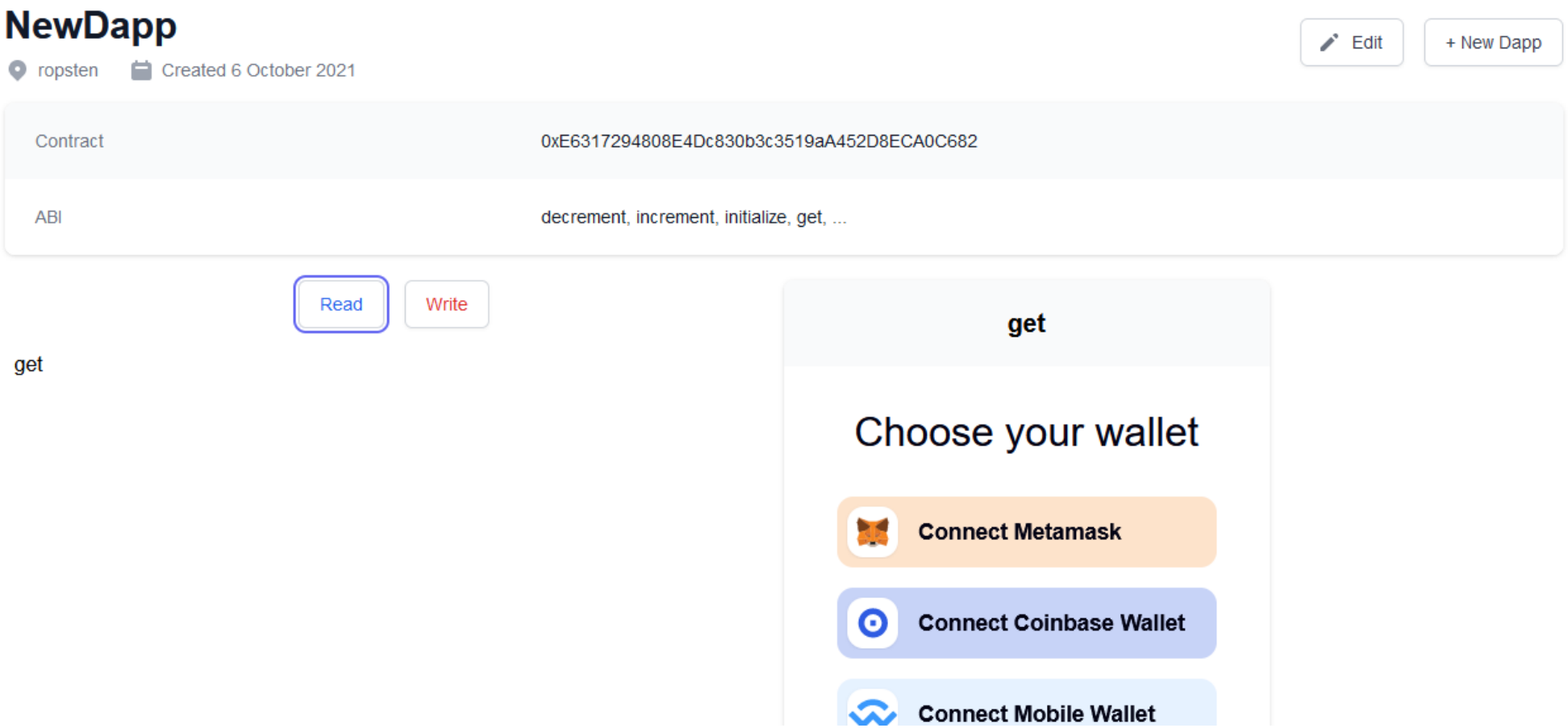
```
contract sample_dapp {
    uint value;
    function initialize (uint x) public {
        value = x;
    }

    function get() view public returns (uint){
        return value;
    }
    function increment (uint n) public {
        value = value + n;
    }
    function decrement   (uint n) public {
        value = value - n;
    }
}
```

# Screenshot of working of DApp




1. Click on initialize and connect with Metamask
2. First initialize the value
3. Check the value in **Read** module under **get** function.

# Screenshot of working of DApp



# Task

- Create a DApp of the smart contract created in the last lab (in practical 5)