NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Sandip Chakraborty
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 12: Smart Contracts and the Permissioned Models of Blockchain

CONCEPTS COVERED

- Smart Contracts and automated code execution
- Permissioned Blockchain

KEYWORDS

- Smart Contracts
- Blockchain 2.0 and 3.0
- Permissioned Models
- Enterprise Blockchains

DLTs for Code Execution

- DLTs can contain information beyond financial transactions
  - What about submitting an executable code as a transaction?

- **Transaction gets executed ==** Your code is getting executed
  - And you have consensus on the code execution !

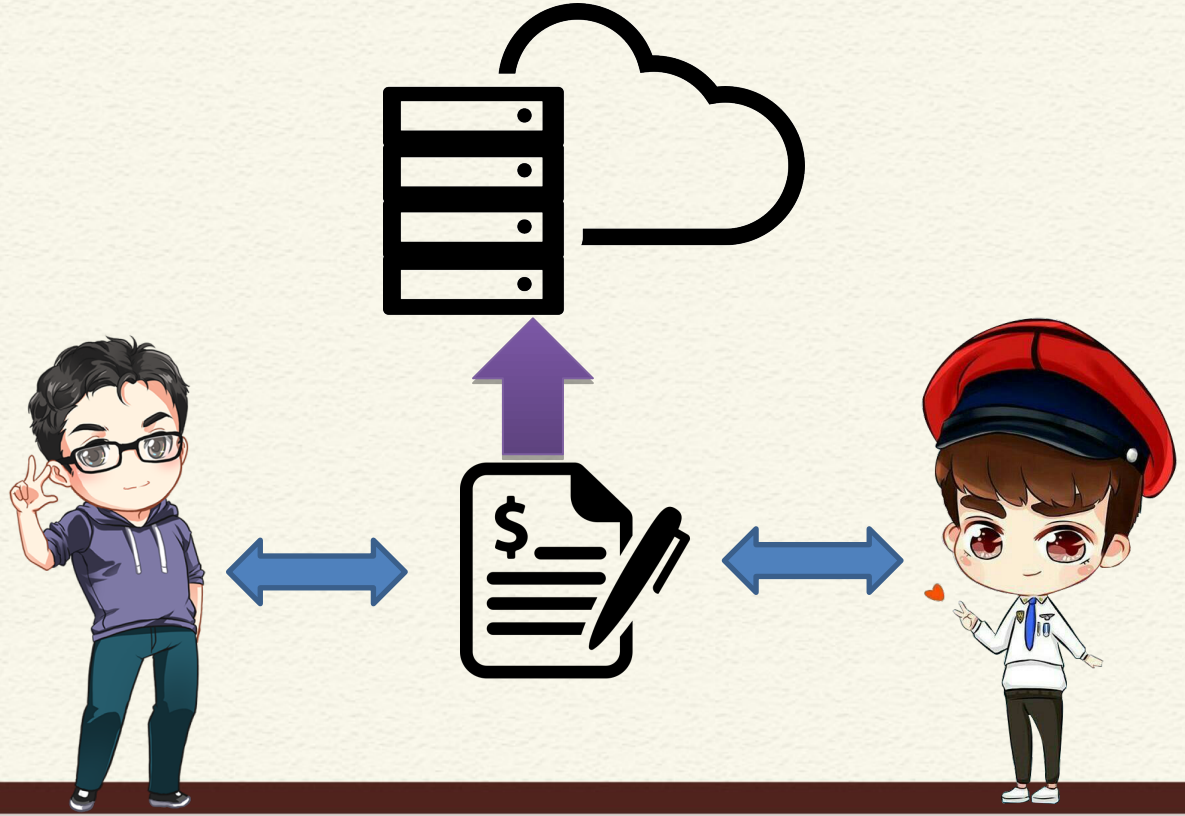
Smart Contracts

Smart Contacts

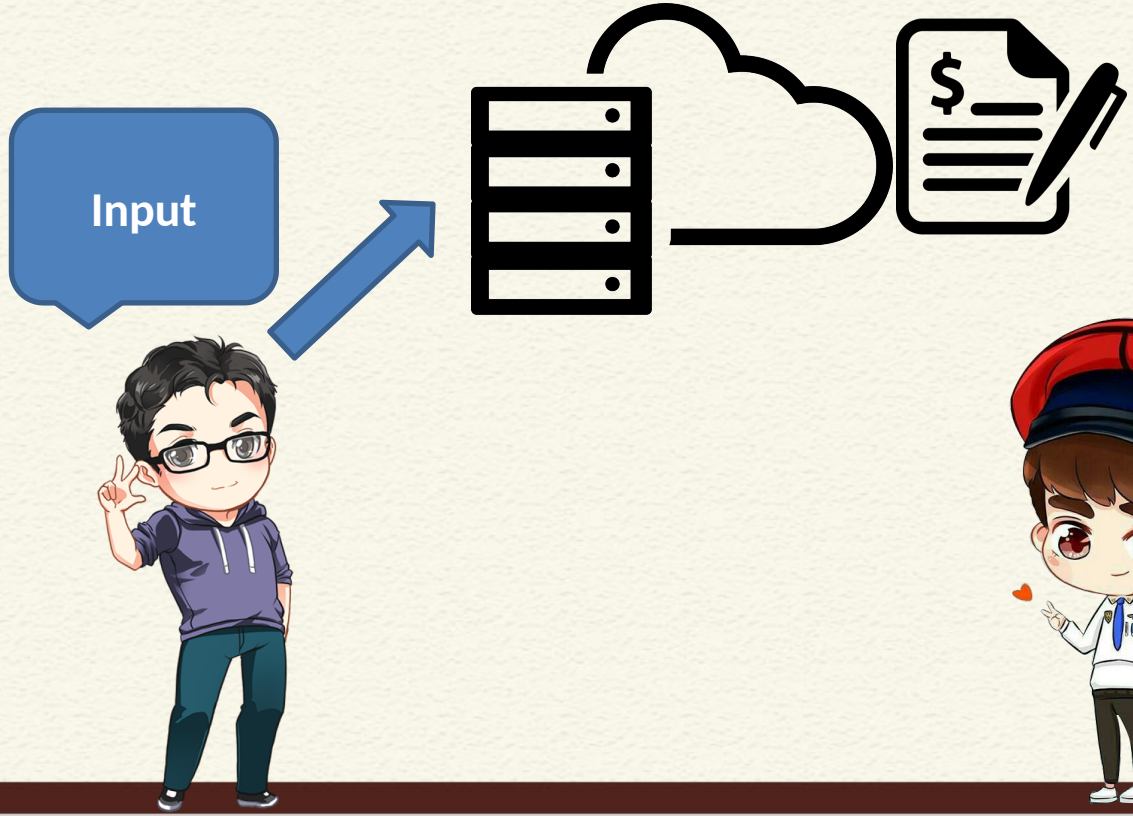
Automatically Execute Code over a Decentralized Platform

Traditional Way of Maintaining Digital Contacts

Traditional Way of Maintaining Digital Contacts



Traditional Way of Maintaining Digital Contacts



Traditional Way of Maintaining Digital Contacts

Traditional Way of Maintaining Digital Contacts

Input

Traditional Way of Maintaining Digital Contacts

Response

Traditional Way of Maintaining Digital Contacts

How will you check that the inputs are valid?

Traditional Way of Maintaining Digital Contacts

What if I deny about an input later on?

Decentralized Code Execution

```
int pay (float *sndAcc, float *rcvAcc, float amount) {
    if (*sndAcc < amount) return -1;
    else {
         *sndAcc -= amount;
         *rcvAcc += amount;
         return 1;
    }
}

int deliverGoods (int count, int pricePerC) {
    int success = pay (sender, receiver, count*pricePerC);
    if(success == 1) {
       sceduleLogistics();
       return 1;
    }
    Return 0;
}
```

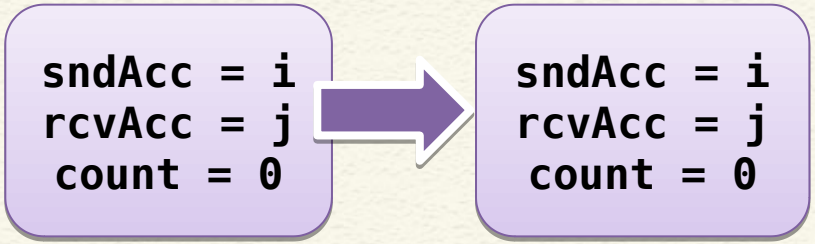
Decentralized Code Execution

```
int pay (float *sndAcc, float *rcvAcc, float amount) {
    if (*sndAcc < amount) return -1;
    else {
         *sndAcc -= amount;
         *rcvAcc += amount;
         return 1;
    }
}

int deliverGoods (int count, int pricePerC) {
    int success = pay (sender, receiver, count*pricePerC);
    if(success == 1) {
       sceduleLogistics();
       return 1;
    }
    Return 0;
}
```

sndAcc = i
rcvAcc = j
count = 0

Decentralized Code Execution

```
int pay (float *sndAcc, float *rcvAcc, float amount) {
    if (*sndAcc < amount) return -1;
    else {
         *sndAcc -= amount;
         *rcvAcc += amount;
         return 1;
    }
}

int deliverGoods (int count, int pricePerC) {
    int success = pay (sender, receiver, count*pricePerC);
    if(success == 1) {
       sceduleLogistics();
       return 1;
    }
    Return 0;
}
```



deliverGoods (10, 4)

Decentralized Code Execution

```
int pay (float *sndAcc, float *rcvAcc, float amount) {
    if (*sndAcc < amount) return -1;
    else {
         *sndAcc -= amount;
         *rcvAcc += amount;
         return 1;
    }
}

int deliverGoods (int count, int pricePerC) {
    int success = pay (sender, receiver, count*pricePerC);
    if(success == 1) {
       sceduleLogistics();
       return 1;
    }
    Return 0;
}
```

deliverGoods (10, 4)
pay(sndAcc, rcvAcc, 40) > 1

sndAcc = i
rcvAcc = j
count = 0

sndAcc = i - 40
rcvAcc = j + 40
count = 40

Decentralized Code Execution

```
int pay (float *sndAcc, float *rcvAcc, float amount) {
    if (*sndAcc < amount) return -1;
    else {
         *sndAcc -= amount;
         *rcvAcc += amount;
         return 1;
    }
}

int deliverGoods (int count, int p.      .C) {
    int success = pay (sender, rece      , count*pricePerC
    if(success == 1) {
       sceduleLogistics();
       return 1;
    }
    Return 0;
}
```

```
deliverGoods (10, 4)
pay(sndAcc, rcvAcc, 40) >
```

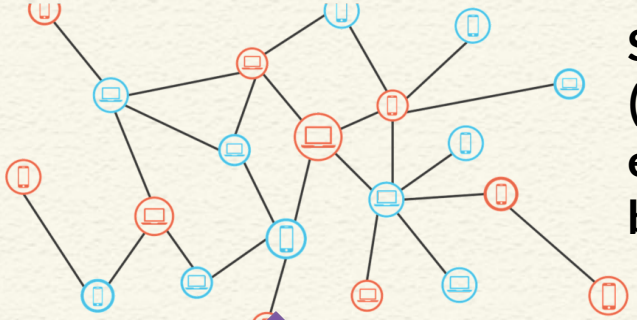
```
sndAcc = i
rcvAcc = j
count = 0
```

```
sndAcc = i - 40
rcvAcc = j + 40
count = 40
```

Put the states of execution in a blockchain

Smart Contract Execution

Smart Contract Execution



Submit the anonymized (through public key encryption) contract to a blockchain network

Smart Contract Execution

Everyone in the network can see and validate the execution steps

Cryptokitties – A Popular Game on Ethereum

Cryptokitties – A Popular Game on Ethereum

Permissioned Model of Blockchain

- PoW (Nakamoto Consensus) works good in an open network
  - But, transaction latency is very high
  - ~10 minutes in Bitcoin block commitment
  - Few seconds to few minutes for Ethereum (depending on the cost that you pay)

Permissioned Model of Blockchain

- PoW (Nakamoto Consensus) works good in an open network
  - But, transaction latency is very high
  - ~10 minutes in Bitcoin block commitment
  - Few seconds to few minutes for Ethereum (depending on the cost that you pay)

- **Can we think of any other Blockchain applications beyond cryptocurrency?**
  - The high latency makes them unsuitable for most of the real-time applications

Permissioned Model of Blockchain

- **Many decentralized applications do not demand an open environment**
  - The food supply chain
  - Know Your Customer (KYC)
  - Trade financing
  - …

Blockchain 3.0

- "**Trustless Decentralization**" over a closed network
  - Automatically transact assets among multiple organizations who do not trust each other
  - Run smart contracts within a consortium of various organizations – the individual organizations know each other but do not trust each other

Blockchain 3.0

- **Advantages:**
  - Go back to the classical distributed consensus protocols – low latency for commitment and high transaction throughput
  - Use "Witness Cosigning" instead of "Proof Mining" for new block generation
    - Classical Distributed Consensus + Digital Signature

Permissioned Blockchain

- The participants are pre-authenticated and pre-authorized
  - But they can still behave maliciously

- Run blockchain (and smart contracts) on top of this closed network
  - Ensure trusted computing among the participants

Conclusion

- Smart Contracts revolutionizes the blockchain/DLT applications
  - Supports automated code execution over a decentralized platform

- Permissioned blockchains have emerged for enterprise applications

Conclusion

- Smart Contracts revolutionizes the blockchain/DLT applications
  - Supports automated code execution over a decentralized platform
- Permissioned blockchains have emerged for enterprise applications
- **Now let us explore the detailed internal structure of a blockchain …. our next lecture**


Thank
you