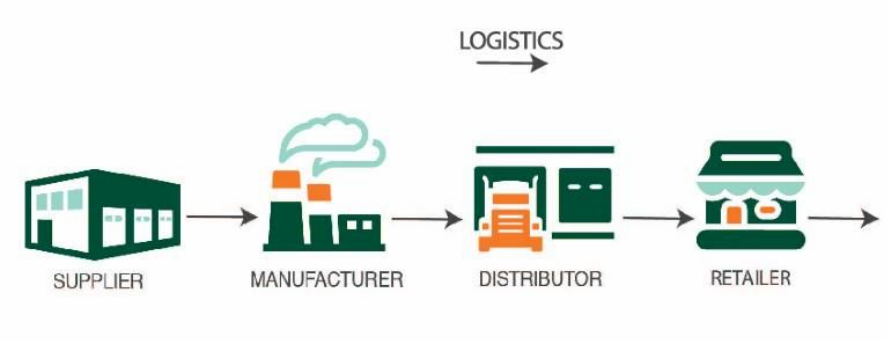
Selected Topics Blockchain Security

Yue Duan

ILLINOIS TECH
College of Computing

Blockchain Basics

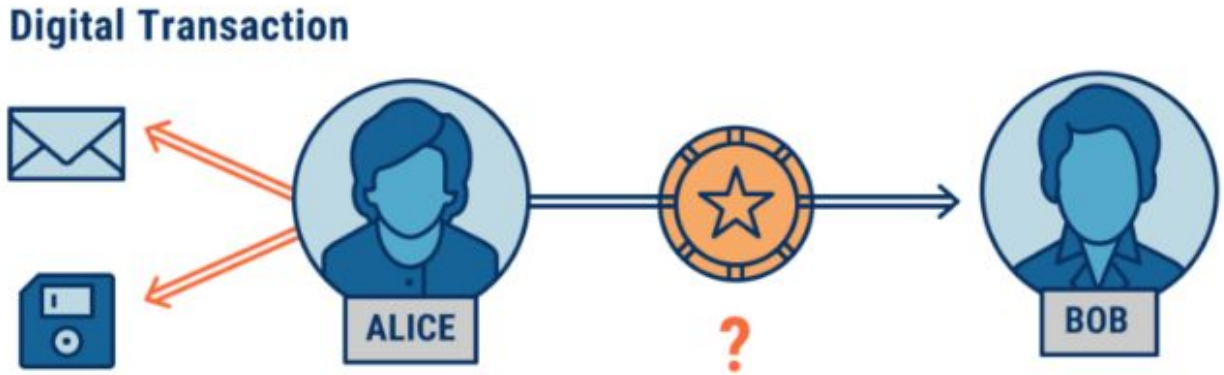


Blockchain Basics

- Scenario 1: Alice hands Bob a token physically
  - Central bank is required

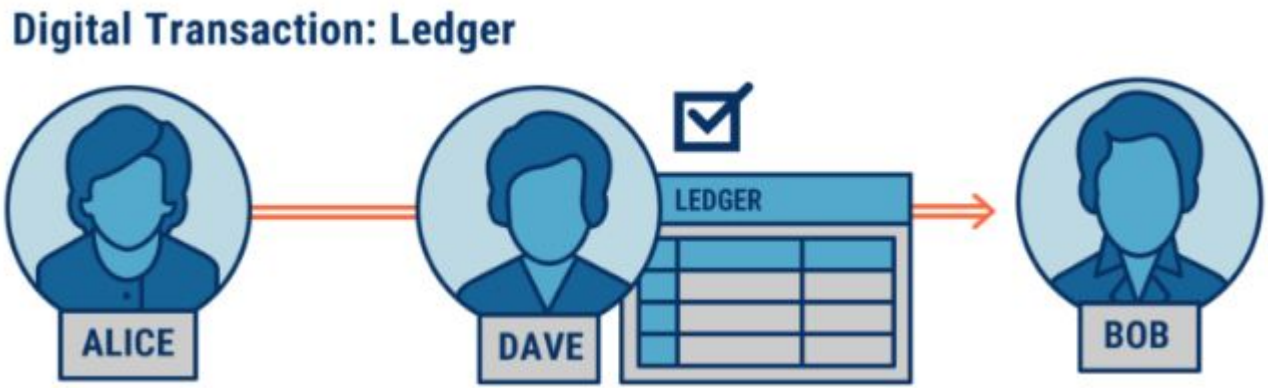
Blockchain Basics

- Scenario 2: Alice hands Bob a digital token via email
  - How to stop Alice from forging a copy of the token?
  - How to stop Alice from using the same token twice?
  - If Alice and Bob own the same token, who is the real owner?



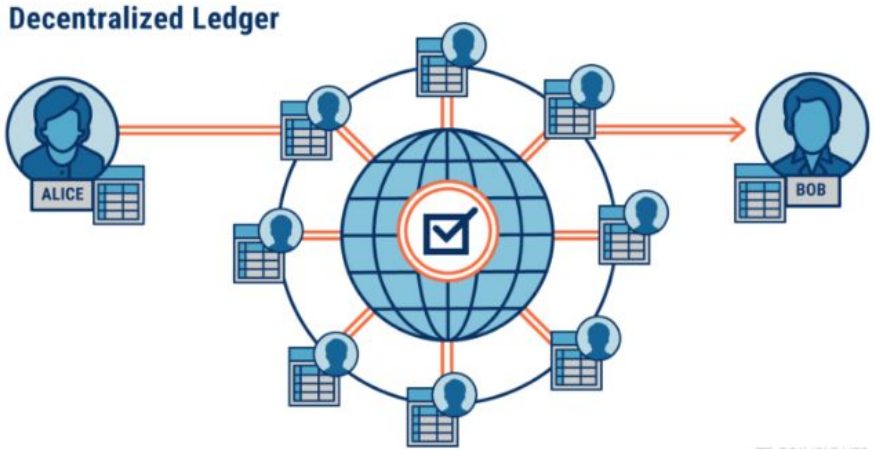
Blockchain Basics

- Scenario 3: Alice hands Bob a digital token via a trusted third-party
  - Can we trust Dave?
    - He may change the ledger
    - He may make mistakes



Blockchain Basics

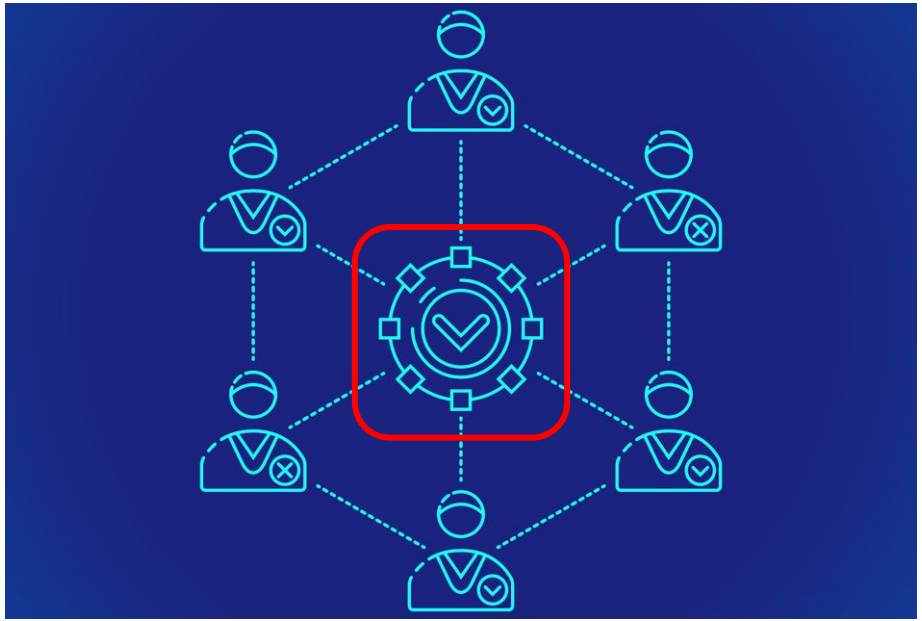
- Scenario 4: Alice hands Bob a digital token using a decentralized ledger
  - Every participant has a copy of ledger
  - only add a transaction to the ledger when majority of participants agree
  - Ledger syncs among all participants



Blockchain Basics

a digital ledger of transactions

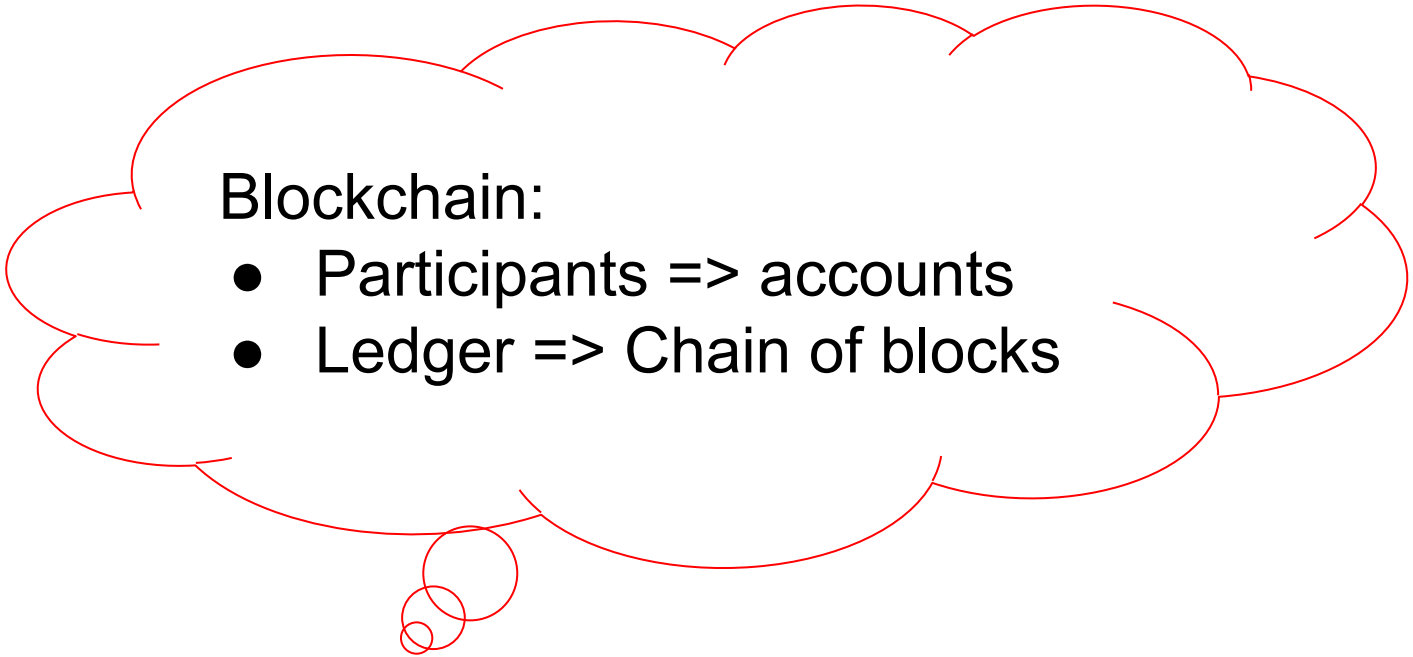
duplicated and distributed across the entire network



decentralized system

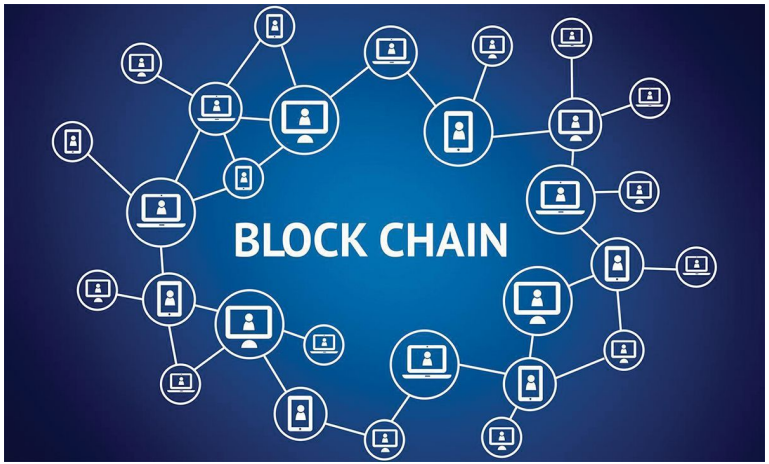
consensus mechanism

Blockchain Basics

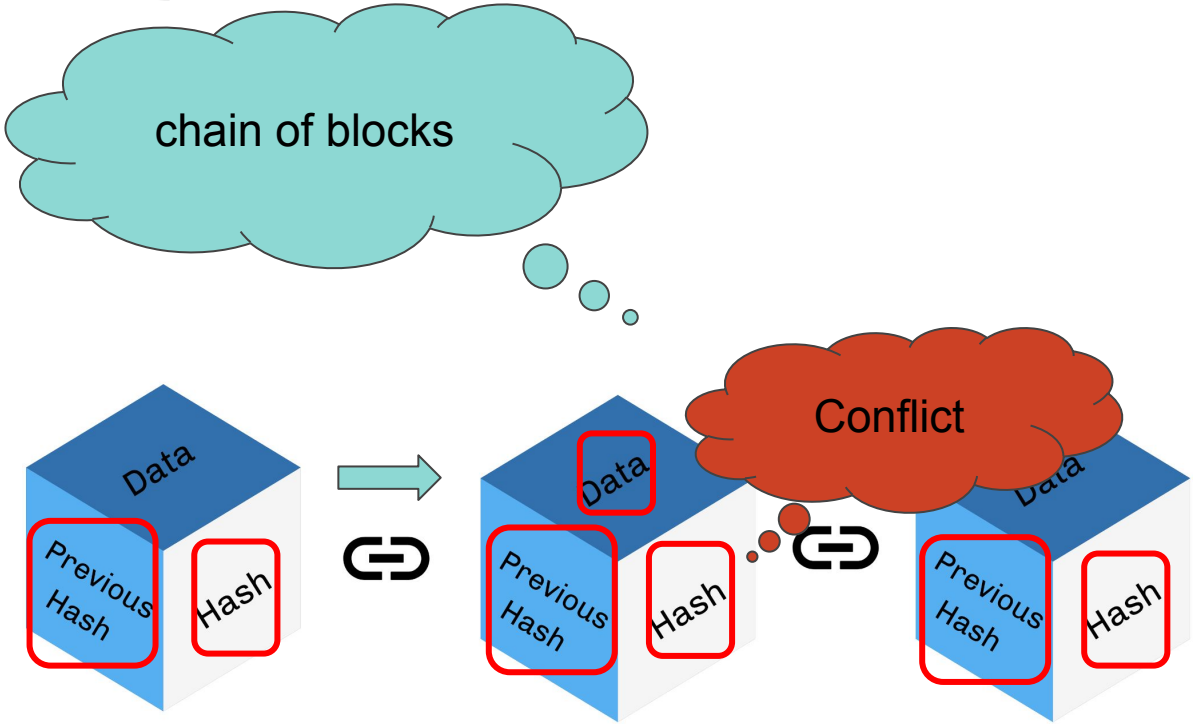


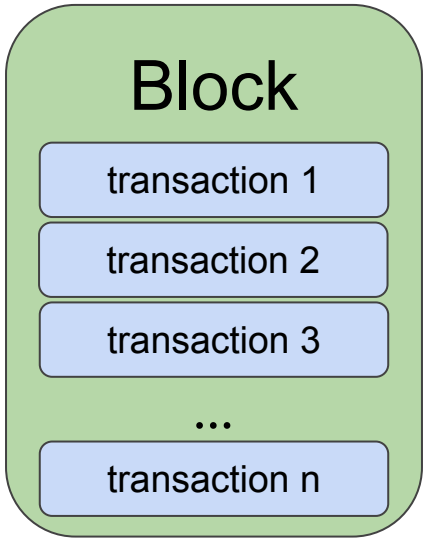
Blockchain Basics

- A peer-2-peer network
  - Decentralized system
  - No central node or central government
  - Two ways of viewing blockchain
    - A ledger with transactions
    - A sequence of state changes



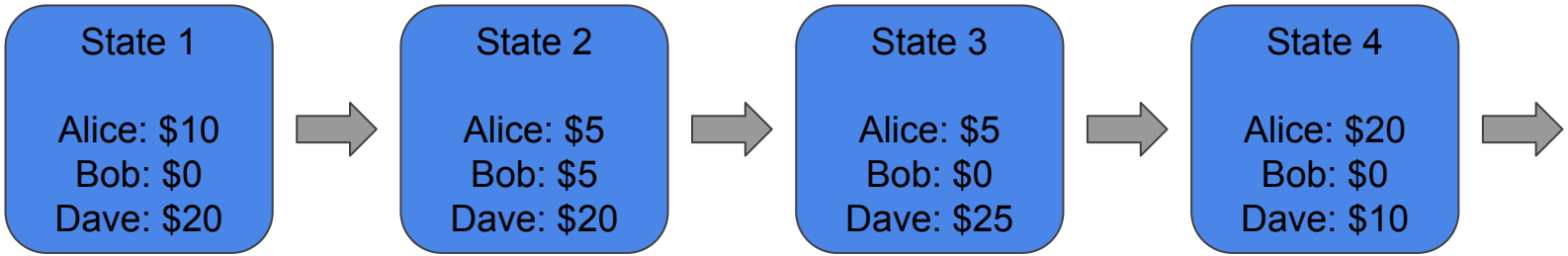
Blockchain Basics





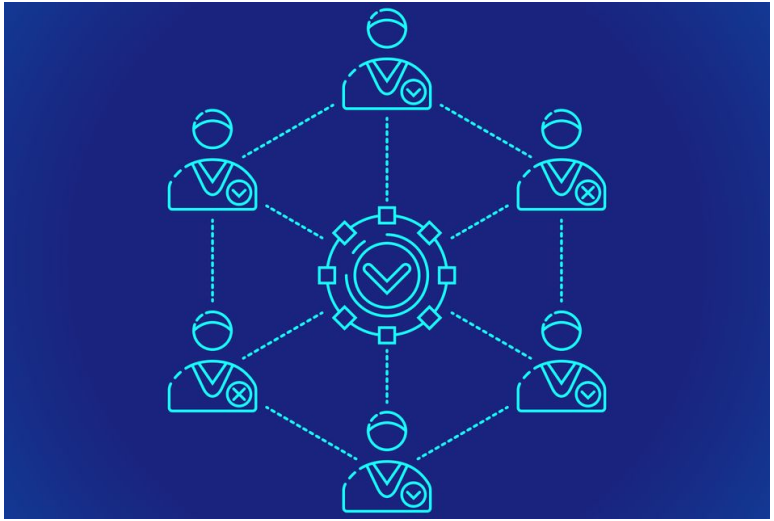
Blockchain Basics

- A sequence of state changes
  - At any given time ***i*** , there is a state ***s***
  - Blockchain is nothing but a sequence of these state changes chained together



Blockchain Basics

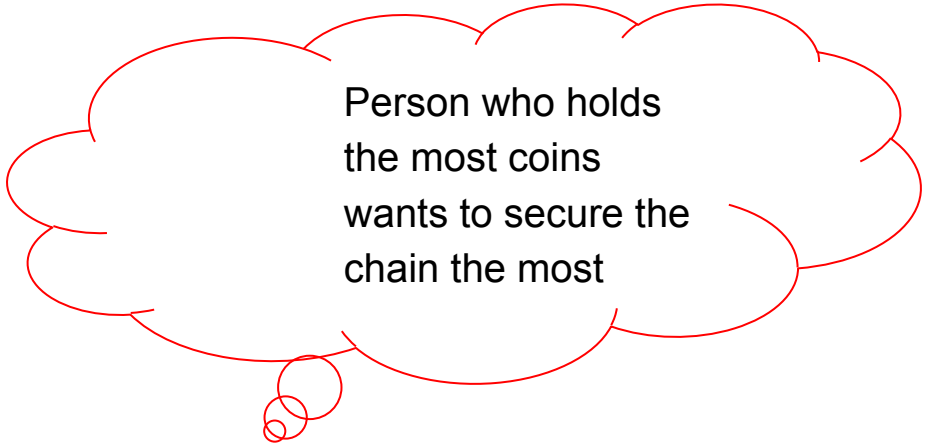
- Consensus mechanisms:
  - How to confirm a transaction in blockchain?
  - Two major consensus protocols
    - Proof of Work (PoW)
      - Most popular
      - Used by bitcoin, ethereum, etc
    - Proof of Stake (PoS)
      - Very promising



Blockchain Basics

- Proof of Work (PoW)
  - Miners solve puzzles to mine blocks (a sequence of transactions)
    - Hard to find a solution
    - Easy to verify the correctness of a solution
  - When a miner finds a solution
    - The new block is broadcast to the network for verification
    - Append the block to the blockchain
  - Limitations:
    - HUGE power consumption
    - 51% attacks

Blockchain Basics



- Proof of Stake (PoS)
  - Like shareholders of a company
  - Participants must have a stake
    - Stake: usually by owning some cryptocurrencies
    - To have a chance of selecting, verifying and validating transactions
  - Factors of having the chance
    - the amount of stake
    - the duration of the stake
  - No mining involved
  - No need for the entire network to be involved in validation process

Blockchain Basics

Transparent

Distributed

Anonymous

Secure

Immutable

Time-stamped

Smart Contract Basics

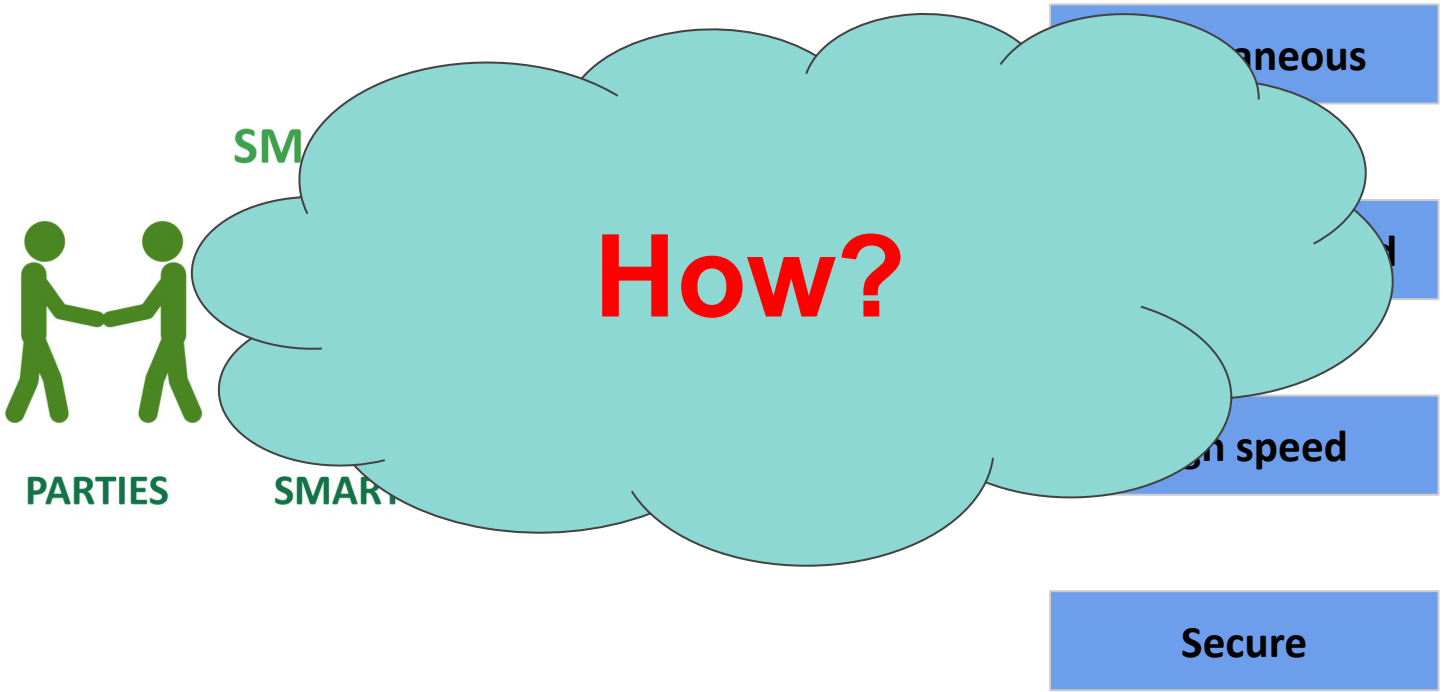
Physical contracts

Trusted 3rd party

Slow speed

High cost

Smart Contract Basics

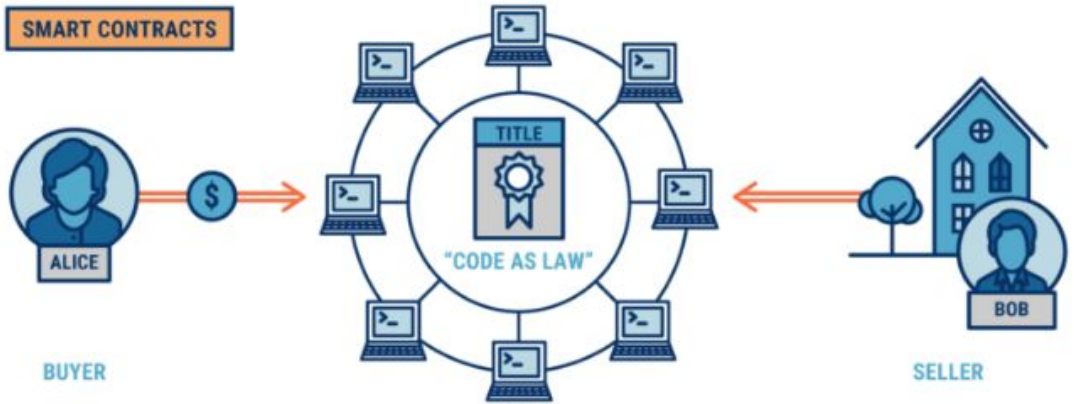


Smart Contract Basics

- Scenario 1: buyer buys a house
  - Traditional contract
  - Third-party involved

Smart Contract Basics

- Scenario 2: buyer buys a house via Ethereum
  - Write code to implement the contract
  - Code is stored in Ethereum blockchain
  - Logic is immutable



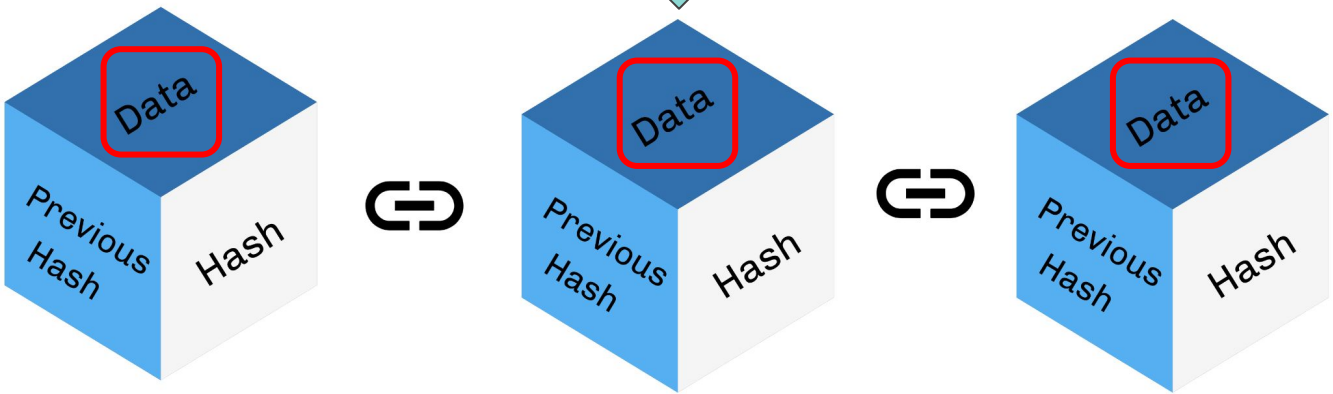
Smart Contract Basics

flexible
immutable
self-executed
high speed

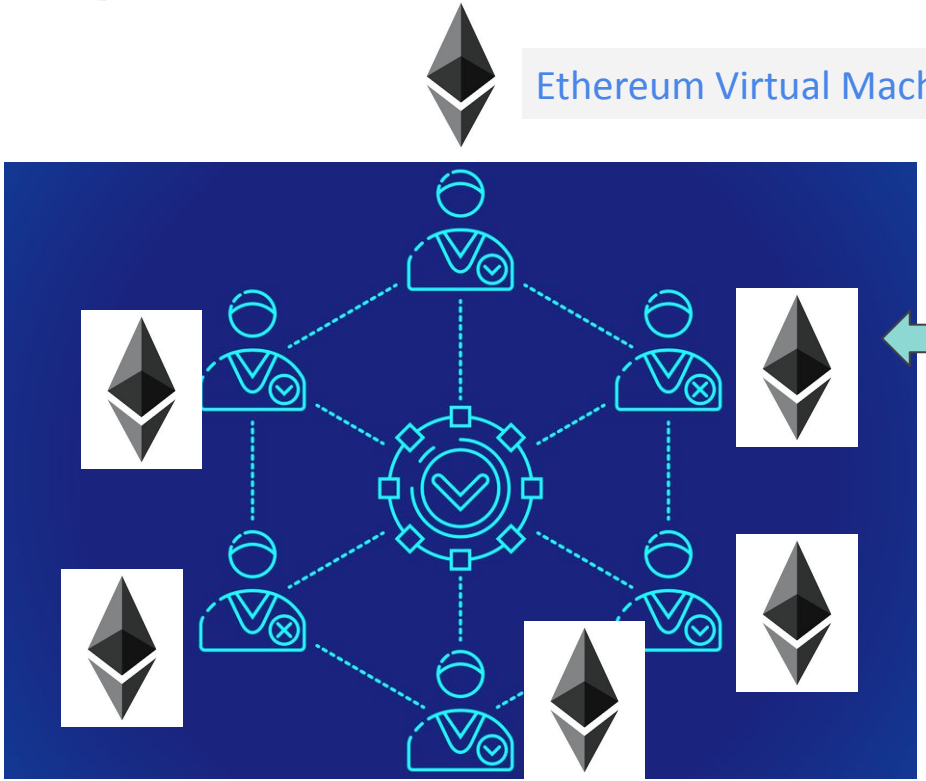
```
/// Give your vote (inclu
/// to proposal `prop
function vote(uint p
    Voter storage s
    require(!sender.
    sender.voted = tru
    sender.vote = pro

    // If `proposal`
    // this will throw
    // changes.
    proposals[proposal].voteCoun
}

/// @dev Computes the winning pro taking all
/// previous votes into account
function winningProposal() pub view
        returns (uint winning proposal_)
{
    uint winningVoteCount = 0;
    for (uint p = 0; p < proposals.length; p++) {
        if (proposals[p].voteCount > winningVoteCount) {
            winningVoteCount = proposals[p].voteCount;
            winningProposal_ = p;
        }
    }
}
```



Smart Contract Basics



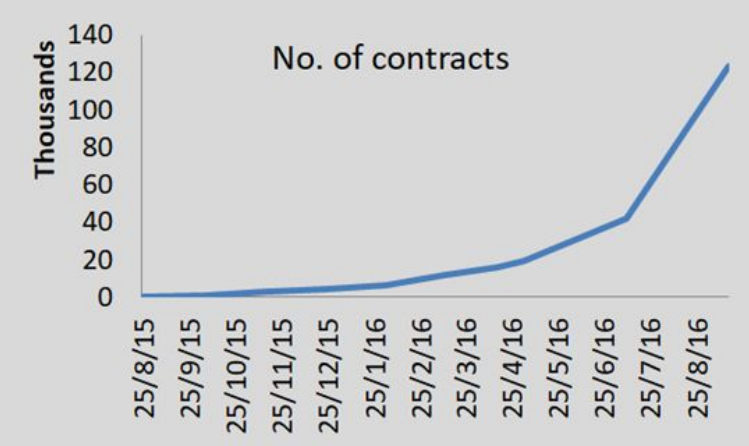
Ethereum Virtual Machine

```
/// Give your vote (including votes delegated to you)
/// to proposal `proposals[proposal].name`.
function vote(uint proposal) public {
    Voter storage sender = voters[msg.sender];
    require(!sender.voted, "Already voted.");
    sender.voted = true;
    sender.vote = proposal;

    // If `proposal` is out of the range of the array,
    // this will throw automatically and revert all
    // changes.
    proposals[proposal].voteCount += sender.weight;
}

/// @dev Computes the winning proposal taking all
/// previous votes into account.
function winningProposal() public view
        returns (uint winningProposal_)
{
    uint winningVoteCount = 0;
    for (uint p = 0; p < proposals.length; p++) {
        if (proposals[p].voteCount > winningVoteCount) {
            winningVoteCount = proposals[p].voteCount;
            winningProposal_ = p;
        }
    }
}
```

Smart Contract Basics



Smart Contracts Market Size to Reach USD 345.4 Million by 2026 at CAGR 18.1% | Valuates Reports

Valuates Reports®

Smart Contract Basics

- Smart contract
  - A program that is running on top of blockchain
  - Code logic is automatically enforced by blockchain
  - No party can ever change the code once it is put in the blockchain

```
/// Give your vote (including votes delegated to you)
/// to proposal `proposals[proposal].name`.
function vote(uint proposal) public {
    Voter storage sender = voters[msg.sender];
    require(!sender.voted, "Already voted.");
    sender.voted = true;
    sender.vote = proposal;

    // If `proposal` is out of the range of the array,
    // this will throw automatically and revert all
    // changes.
    proposals[proposal].voteCount += sender.weight;
}

/// @dev Computes the winning proposal taking all
/// previous votes into account.
function winningProposal() public view
        returns (uint winningProposal_)
{
    uint winningVoteCount = 0;
    for (uint p = 0; p < proposals.length; p++) {
        if (proposals[p].voteCount > winningVoteCount) {
            winningVoteCount = proposals[p].voteCount;
            winningProposal_ = p;
        }
    }
}
```

Blockchain Security issues

- Double-spending attack
  - a.k.a, 51% attack
  - Not as impractical as many people would expect
  - Caused by deep chain reorganization

“If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains.”

--- bitcoin white paper

PoW 51% Attack Cost

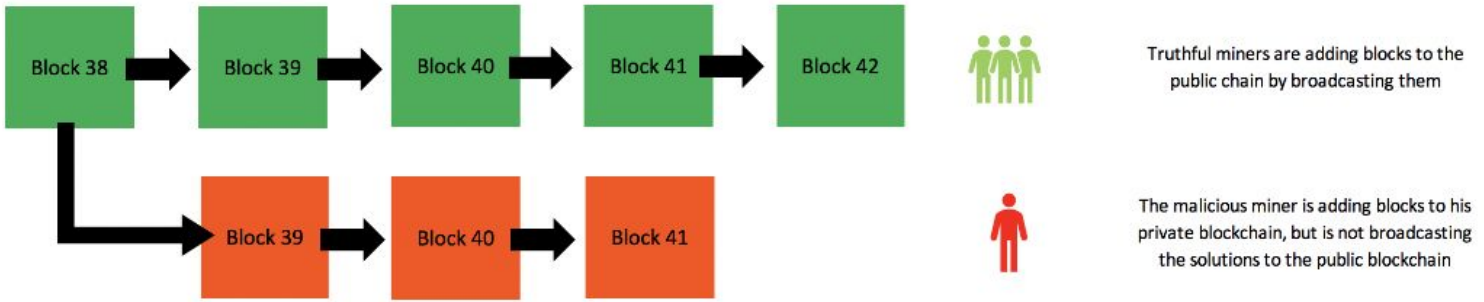
This is a collection of coins and the theoretical cost of a 51% attack on each network.

Learn More | ⚡ Tip

Name	Symbol	Market Cap	Algorithm	Hash Rate	1h Attack Cost	NiceHash-able
Bitcoin	BTC	$185.00 B	SHA-256	136,198 PH/s	*$565,959*	0%
Ethereum	ETH	$37.52 B	Ethash	232 TH/s	*$276,331*	4%
BitcoinCashABC	BCH	$4.06 B	SHA-256	2,480 PH/s	*$10,305*	18%
BitcoinSV	BSV	$3.10 B	SHA-256	2,144 PH/s	*$8,910*	21%
Litecoin	LTC	$3.07 B	Scrypt	314 TH/s	*$19,380*	4%
Dash	DASH	$723.90 M	X11	7 PH/s	*$2,643*	2%
Zcash	ZEC	$579.78 M	Equihash	7 GH/s	*$14,174*	3%
EthereumClassic	ETC	$575.68 M	Ethash	3 TH/s	$3,865	280%
BitcoinGold	BTG	$142.71 M	Zhash	941 KH/s	*$283*	50%
Ravencoin	RVN	$112.15 M	KawPow	2 TH/s	*$4,427*	34%

Double Spend Attack Overview

- When a miner finds a solution, it is supposed to be broadcasted to all other miners.
- However, a malicious miner can create an offspring of the blockchain by not broadcasting the solutions of his blocks to the rest of the network

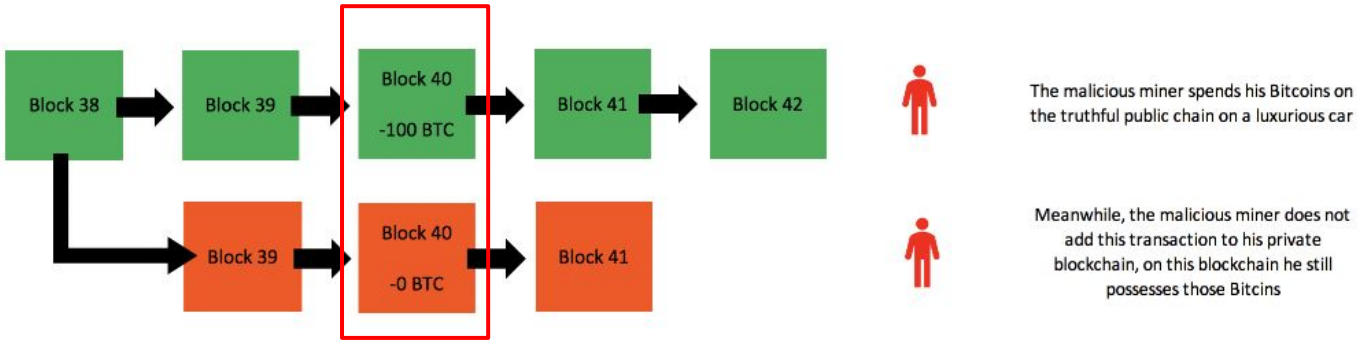


There are now two versions of the blockchain. The red blockchain can be considered in 'stealth' mode.

Source: https://medium.com/coinmonks/what-is-a-51-attack-or-double-spend-attack-aa108db63474

Double Spend Attack Overview

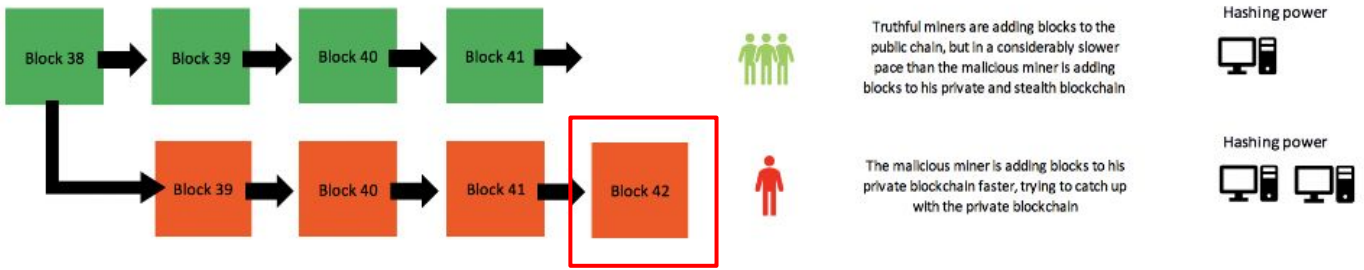
- The malicious miner then spends cryptocurrency on public chain.
- This transaction, however, is not shown in his private chain.



The corrupt miner excludes his own transaction in his private blockchain.

Double Spend Attack Overview

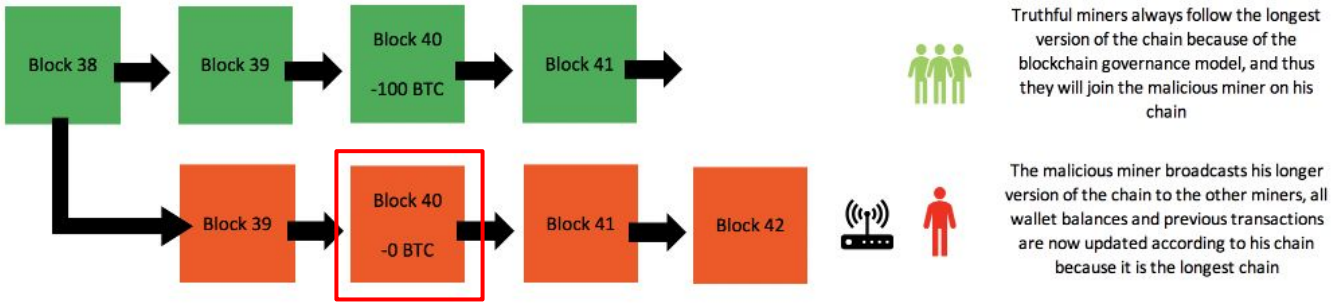
- The malicious miner now tries to add more blocks in his private chain than the public chain.



The corrupt miner is now adding blocks to his private chain faster because he has more hashing power.

Double Spend Attack Overview

- Eventually, malicious miner broadcasts his longer version of the chain, rendering his previous transaction reversed.



The corrupt miner broadcasts its chain to the rest of the network once it is longer (heavier) than the original chain.

Reentrancy attack

- Key idea: find a function that makes an external call to another contract before it resolves any effects

```
function withdraw(uint amount) public {
 ① if (credit[msg.sender] >= amount) {check
 ②   msg.sender.call.value(amount)(); transfer
 ③   credit[msg.sender] -= amount;  update
  }
}
```

Reentrancy attack

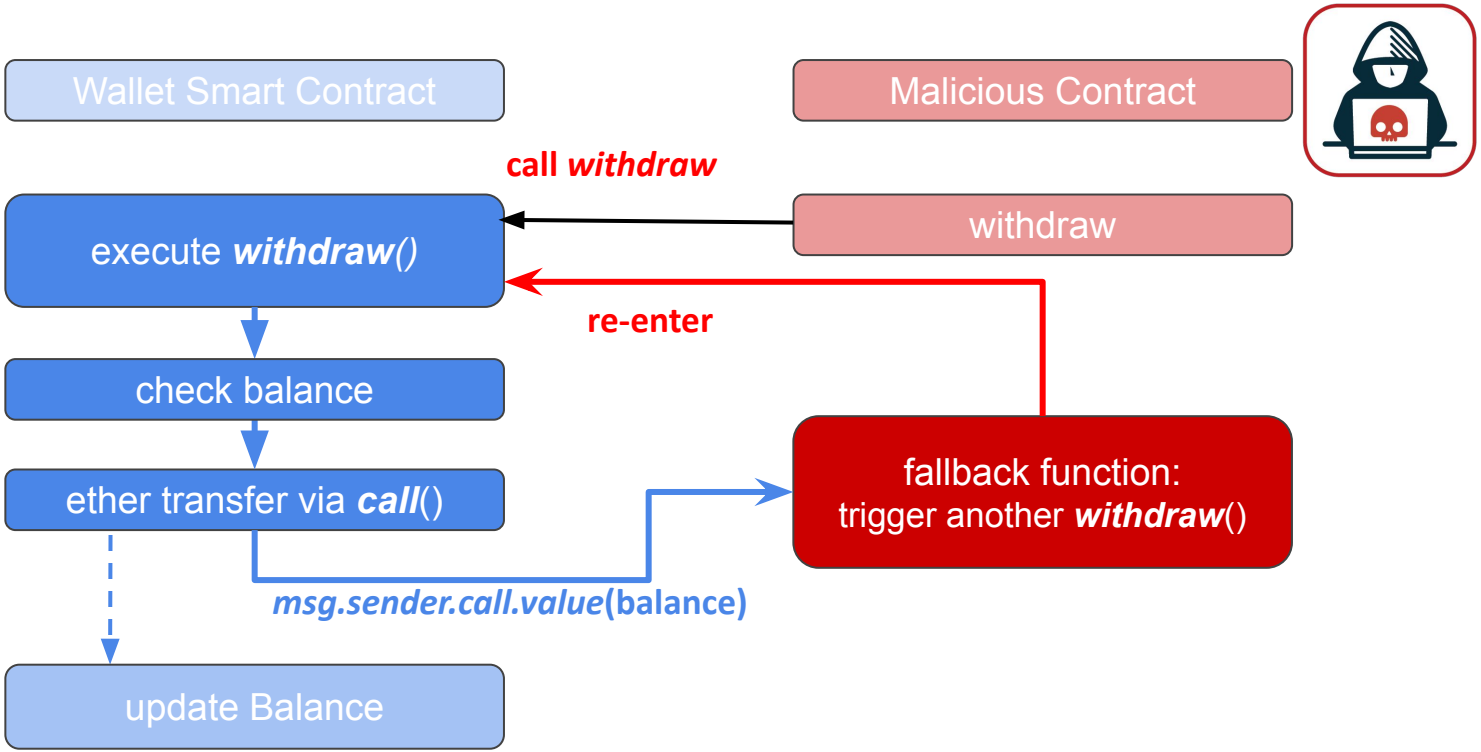
- fallback function:
  - a function without name or parameter
  - can be overridden by developers
  - executed when ether is transferred to the contract

```
contract EveryContract {

  function () public {
  }
}
```

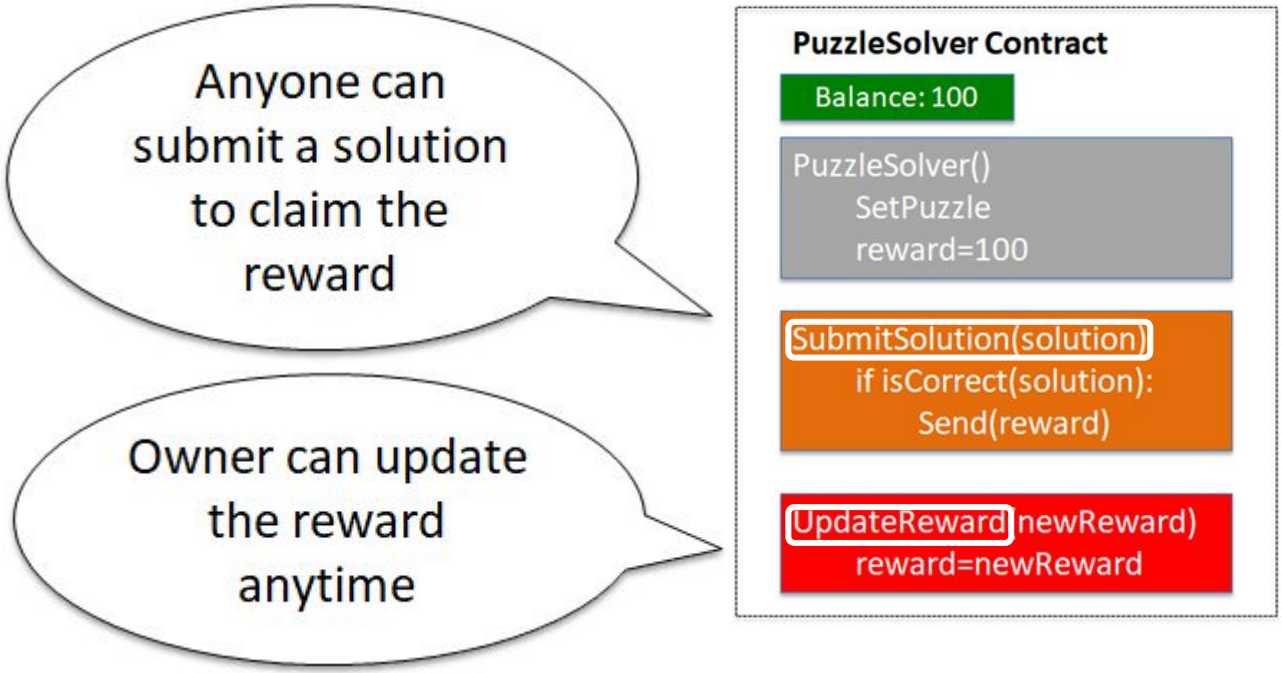
```
function () public payable {
    if (address(this).balance < 999999 ether) {
        callWithdrawBalance(msg.sender);
    }
}
```

Reentrancy attack



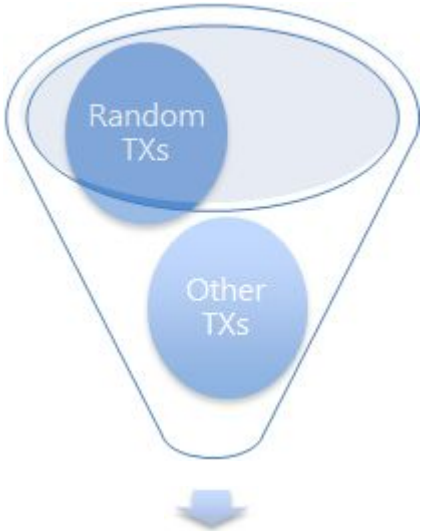
Transaction Ordering Dependence

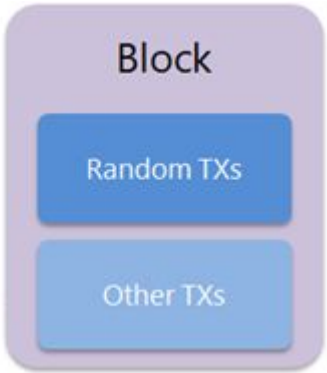
- Key idea: transaction order is not deterministic



Transaction Ordering Dependence

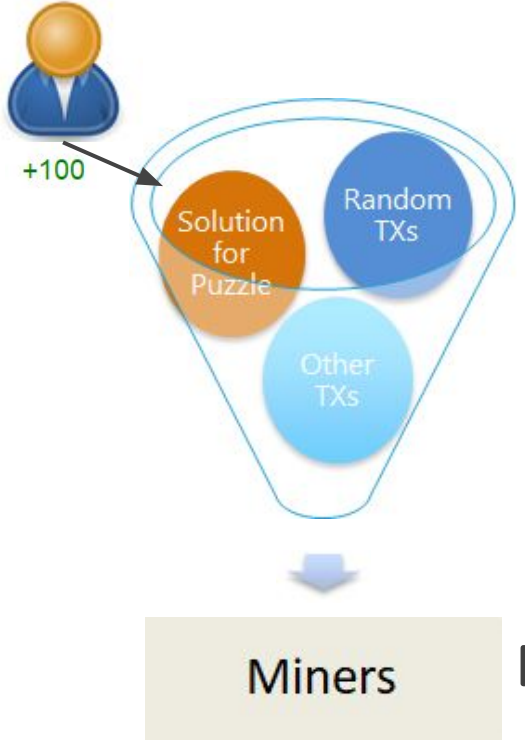
Scenario 1





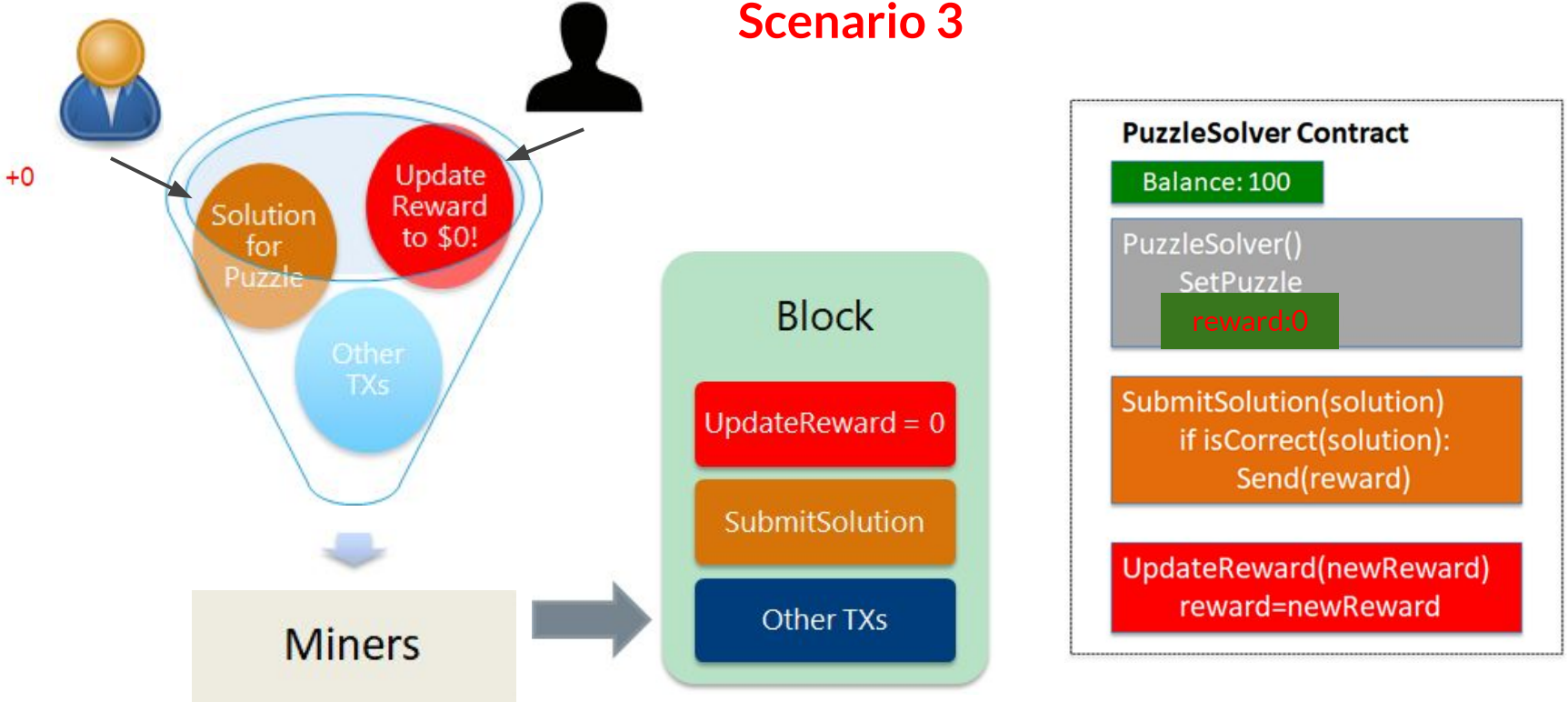
Transaction Ordering Dependence

Scenario 2



Transaction Ordering Dependence

Scenario 3



Thank you!

Question?