BLOCKCHAIN

ETHEREUM

Welcome at SfeirSchool
Blockchain 200


DÉBUTANT
CONFIRMÉ
EXPERT
School

1. **Ethereum**: Introduction & Concepts

1. Ethereum: Introduction & Concepts

CORE CONCEPTS

- ***Application***
- ***Smart Contract***
- ***Ether And Gas***
- ***Ethereum Wallet***
- ***Consensus***

1. Ethereum: Introduction & Concepts

CORE CONCEPTS

Application

- *DAPP*
  - *Backend <-> Smart Contract*
  - *Frontend*

1. Ethereum: Introduction & Concepts

CORE CONCEPTS

Smart Contract

- *Software code*
- *Maps physical contracts to digital world*
- *Run inside an Ethereum virtual machine (EVM)*

1. Ethereum: Introduction & Concepts

CORE CONCEPTS

(Smart contracts …)

- *… are written in using solidity*
- *Identified by a **unique address** (Pk)*
- *Methods of smart contract **can be invoked from a transaction***

1. Ethereum: Introduction & Concepts

CORE CONCEPTS

(Smart contracts ...)

- *Different language*
  - *Solidity*
  - *Serpent*
  - *LLL*

1. Ethereum: Introduction & Concepts

CORE CONCEPTS

Ether And Gas

- *Ether = digital fuel for running a smart contract*
- *Gas = computation work*

1. Ethereum: Introduction & Concepts

CORE CONCEPTS

Ethereum Wallet

- *Hold and secure ETHER and other assets*
- *Allow to deploy, run and use smart contract*

1. Ethereum: Introduction & Concepts

CORE CONCEPTS

Consensus

- *(Proof of work)*
- *Proof of stack*
- *Two kind of nodes:*
  - *regular nodes*
  - *miners*

2. **Ethereum**: Transactions

2. Ethereum: Transaction

Transaction

- *“Signed data package to transfer ETHER from account to another account”*

2. Ethereum: Transaction

Transaction

- *Transaction can:*
  - ***Invokes** methods of a contract*
  - ***Deploy** a new contract*

2. **Ethereum**: Transaction

Transaction

- *Transaction contains:*
  - ***Recipient***
  - ***Signature*** *identifying the sender*
  - ***Amount*** *of Ether to transfer*

2. Ethereum: Transaction

Transaction

- *(...):*
  - ***Gas Limit*** *- transaction execution is allowed to take*
  - ***Gas Price*** *- cost of transaction*
  - ***Transaction fees*** *(Gas Price * Gas used)*

3. **Ethereum**: Accounts

3. Ethereum: accounts

Accounts

- *Need asymmetric key pairs (to create account)*
- *Cryptographic Algorithm ECC(Elliptic Curve Cryptographic)*

3. Ethereum: accounts

Accounts

- *Need asymmetric key pairs (to create account)*
- *Cryptographic Algorithm ECC(Elliptic Curve Cryptographic)*

4. Assets

4. Ethereum: assets

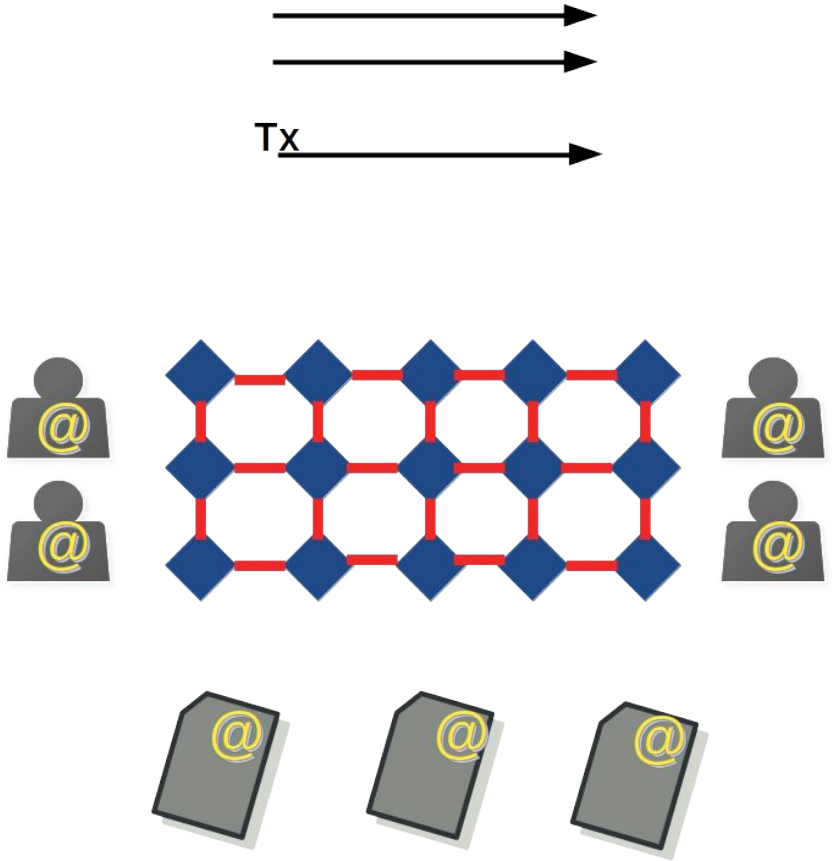
Assets

- *ETHER = crypto currency*
- *TOKEN*
  - *Usage token (ex. Golem token)*
  - *Work token (Identify, Shareholder)*
  - *ERC20 / ERC223 / ERC777 / ERC827*

5. Ethereum: Sum Up

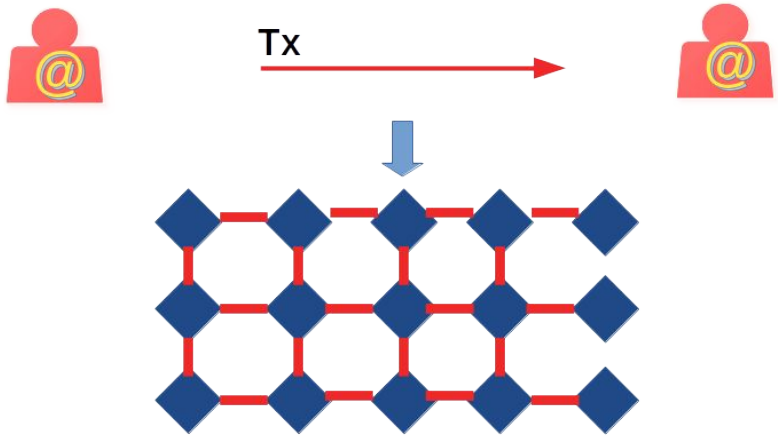
5. Ethereum: Sum-up

sum-up



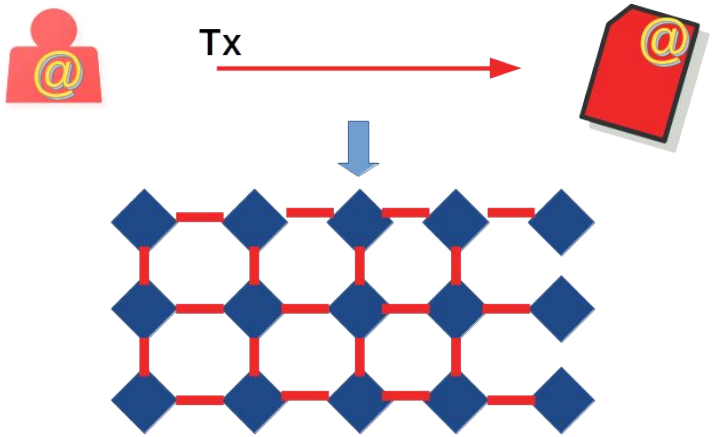
5. Ethereum: Sum-up

sum-up



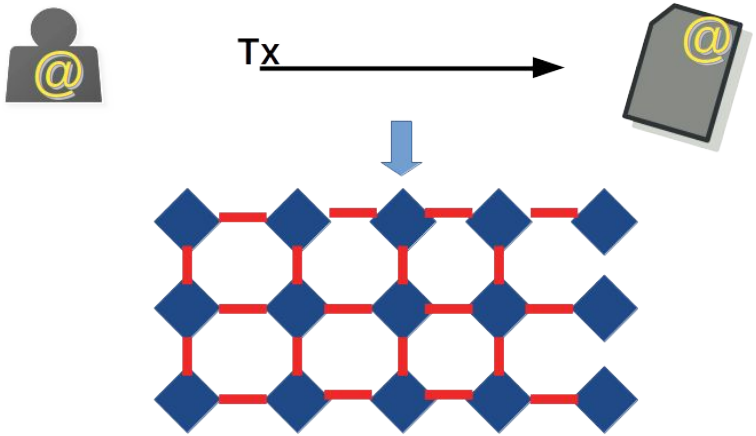
5. Ethereum: Sum-up

sum-up



5. Ethereum: Sum-up

sum-up



ETHER AS FUEL

GAS (PRICE)

Ethereum: Sum-up

- Permissionless:
  - Dap (based on smart contracts)
  - Multi-language (solidicy, serpent ...)
  - Public/Private
  - Ether, Gas
- Transaction
- Accounts

5. Ethereum: Practice

5. Ethereum: Getting start

Tooling

https://github.com/ConsenSys/ethereum-developer-tools-list

5. Ethereum: Getting start

Tooling

- *DApp client:* ***Geth***
- *Development framework:* ***Truffle***
- *Personal Ethereum blockchain:* ***Ganache***
- *DApp browser:* ***Metamask***
- *Smart contract Language:* ***Solidity, Remix Ide***

GETH

5. Ethereum: Getting start

GETH

Connect on Ethereum platform:

- *Geth to connect on Ethereum*
- *Geth to connect on test space*
- *Geth to setup a private network*

5. Ethereum: Getting start

Ethereum

Connect on Ethereum platform:

- *Geth :* ***https://geth.ethereum.org/***

```
geth --networkid 999 --ipcpath ~/Library/Ethereum/geth.ipc --rpc --rpcaddr
"127.0.0.1" --rpcapi="db,eth,net,web3,personal,web3" --rpcport "8545" --
datadir=./data  --rpccorsdomain "*" console
```

5. Ethereum: Getting start

Ethereum

```
geth --networkid 999 --ipcpath ~/Library/Ethereum/geth.ipc --rpc --rpcaddr
"127.0.0.1" --rpcapi="db,eth,net,web3,personal,web3" --rpcport "8545" --
datadir=./data  --rpccorsdomain "*" console
```

- networkid – The networkid value of 999 signifies a local environment. There are standard predefined networkid values from 1 to 5, like 1 for Frontier that connects to an actual Etherum network, 3 is for Ropsten, which is a Test Ethereum network. Any value other than these predefined values implies a local instance.
- *ipcpath* – This is the IPC endpoint file. IPC or inter process communication allows local processes to communicate with geth using the IPC endpoint.
- *rpc* - Enable remote procedure call of Geth APIs over HTTP JSON-RPC protocol.

5. Ethereum: Getting start

Ethereum

```
geth --networkid 999 --ipcpath ~/Library/Ethereum/geth.ipc --rpc --rpcaddr
"127.0.0.1" --rpcapi="db,eth,net,web3,personal,web3" --rpcport "8545" --
datadir=./data  --rpccorsdomain "*" console
```

- *rpcaddr* and *rpcport* - Specify RPC address and RPC port
- *rpcapi* – List of Geth APIs that would be enabled over RPC port
- *datadir* – The data directory for the databases.
- *rpccorsdomain* - Comma-separated list of domains from which to accept cross-origin requests from the browser. A value of "*" implies accept a request from all domains. Our web application will use XMLHttpRequest to interact with Ethereum node using RPC protocol.
- *console* – This would start the geth node instance and open the console.

TRUFFLE & GANACHE

5. Ethereum: Getting start

Truffles suites

EX.1

- *Objective:*
  - *Start Ganache / Truffle docker images*
  - *Deploy smart contract*
  - *Use smart contract*

METAMASK

5. Ethereum: Getting start

Metamask

Configuration

- *Install Chrome Metamask plugin*
- *Setup plugin*
- *Configure account*

5. Ethereum: Getting start

Truffles suites

EX.2

- *Objective:*
  - *Install metamask*
  - *Configure metamask*

SOLIDITY & REMIX IDE

5. Ethereum: Getting start

Solidity

IFTTT : "IF THIS THEN THAT" logic

- ***If** the first set of instructions are done*
- ***Then** execute the next function*
- *(after) **That** the next and keep on repeating until you reach the end of the contract.*

5. Ethereum: Getting start

Solidity

- ***Source file***
  - *“Pragma version”: Specify certain conditions under which the source file can or cannot run.*
  - *Will be used by a compiler*

ex. ***pragma*** *solidity ^0.4.22;*

5. Ethereum: Getting start

Solidity

- ***Structure of contract***
  - *State variables*

    [type] <reference name>

    ex. ***uint*** *storedData****;***

5. Ethereum: Getting start

Solidity

- ***Type***
  - *Boolean, int/uint, fixed/ufixed,*
  - *address(member, balance, send, …)*

ex.

```
address x = 0x123;
address myAddress = this;
if (x.balance < 10 && myAddress.balance >= 10)
x.transfer(10);
```

5. Ethereum: Getting start

Solidity

- ***Functions***
  - *Functions are the executable units of code within a contract.*

Ex.

```
function bid() public payable {
        // Function
        // ...
}
```

5. Ethereum: Getting start

Solidity

- ***Events***
  - *Events are convenience interfaces with the EVM logging facilities.*

Ex.

```
event SimpleEvent(address bidder, uint amount); // Event
...
emit SimpleEvent(msg.sender, msg.value); // Triggering event
```

5. Ethereum: Getting start

Truffles suites

EX.3

- *Objective:*
  - *Develop a smart contract*
  - *Deploy and run smart contract with remix-ide*

Questions ?