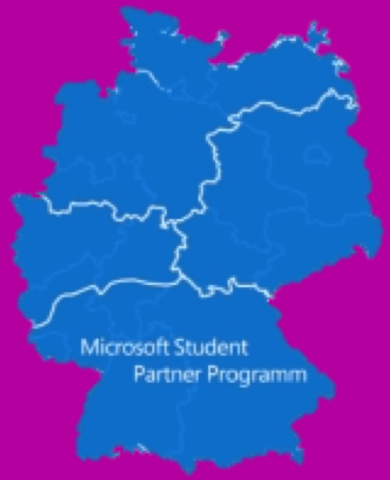
Microsoft Student Partners

Microsoft Student Partners are the future of technology!

Applied Ethereum:

Blockchain & Smart Contracts



StudentPartners.de

01001101
01010011
01010000

CODE IS LIFE

http://blogs.msdn.microsoft.com/StudentPartner

Microsoft

Agenda

Agenda

01001101
01010011
01010000

Time	Agenda Items
	General Introduction to the Blockchain
	Setup Infrastructure
	Ethereum & Smart Contracts
	Smart Contract Use Cases
	Build your own Smart Contract!
	Present your contract to your peers!

Files for this workshop

01001101
01010011
01010000

Please visit:

https://goo.gl/kjNdsc

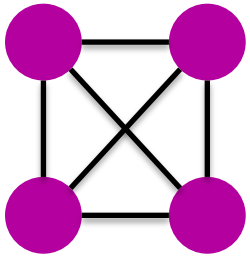
Blockchain

What is a Blockchain?

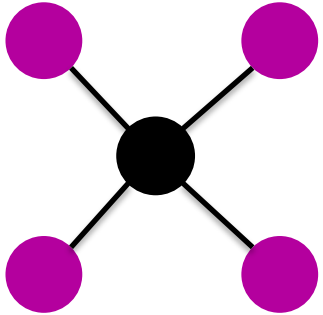
01001101
01010011
01010000

- Transnational Database
  - But: **Everyone** can read entries
- Append-only data structure
- **Peer-to-Peer** network
- Transactions stored in **Blocks**

Peer-to-peer vs. centralized architecture

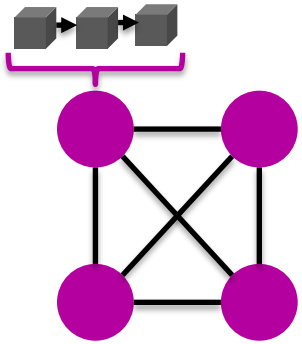


Peer-to-peer network



Centralized network

Peer-to-peer vs. centralized architecture



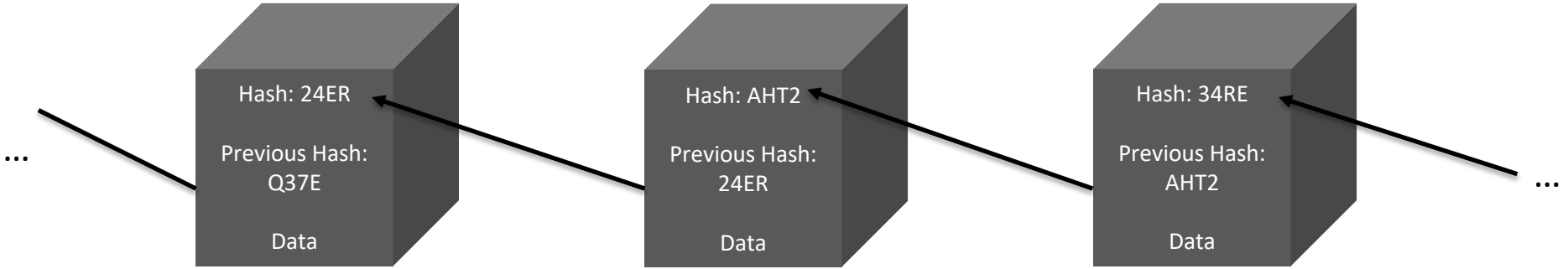
Peer-to-peer network

- Each (full) node stores the whole transactional database (blockchain)

Blocks

01001101
01010011
01010000

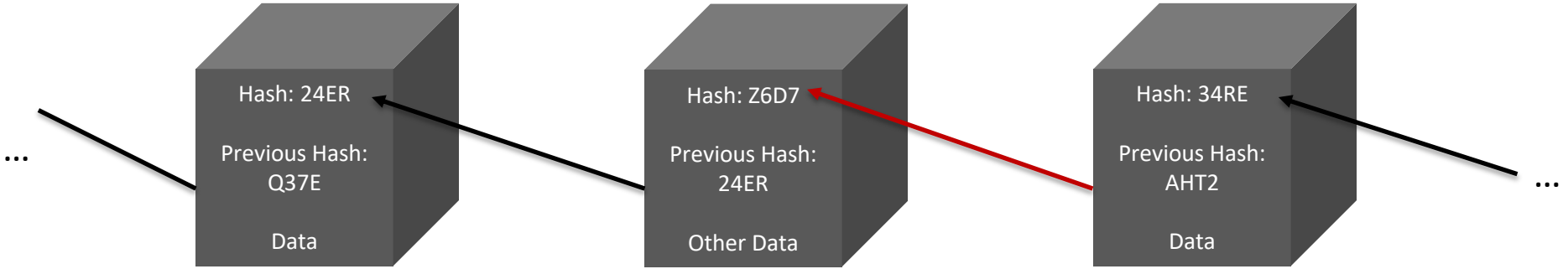
What happens if one block is manipulated?



Blocks

01001101
01010011
01010000

What happens if one block is manipulated?



Full Node vs. Light Node

- Full Node:
  - Stores full blockchain (Bitcoin blockchain: 158.7 Gb)
  - Is connected to other full nodes
    - Miner: Additionally able to create new blocks
- Light Node:
  - Only validates single transactions

Adding new transactions

1. Wallet owner creates and signs transaction
2. Transaction is sent to a full node
3. Transaction is spread through the network
4. Miner builds transaction into block
5. Other nodes validate block and add it to the blockchain

Blockchain Use Cases

01001101
01010011
01010000

ethereum

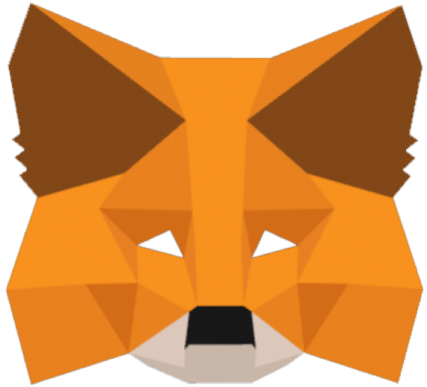
Required Infrastructure

Getting started

01001101
01010011
01010000

What are we using?

- Metamask Ethereum browser extension

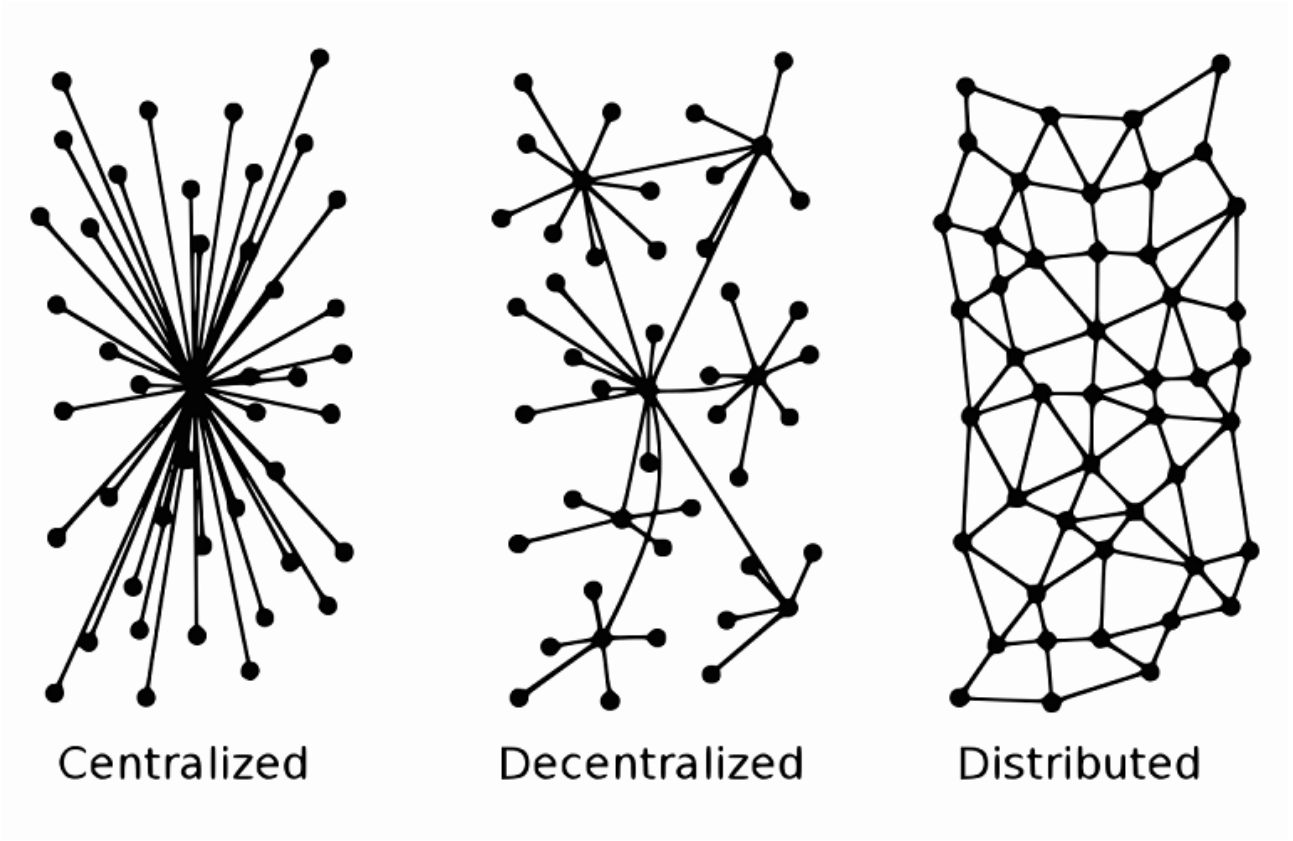


- Remix IDE

remix

- Ropsten Test Network

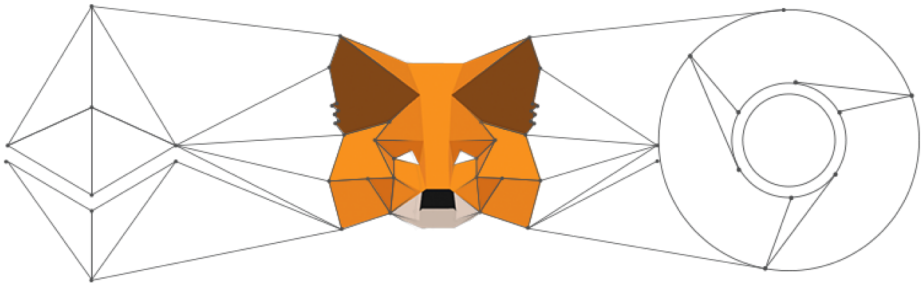
Centralized vs Decentalized



Metamask

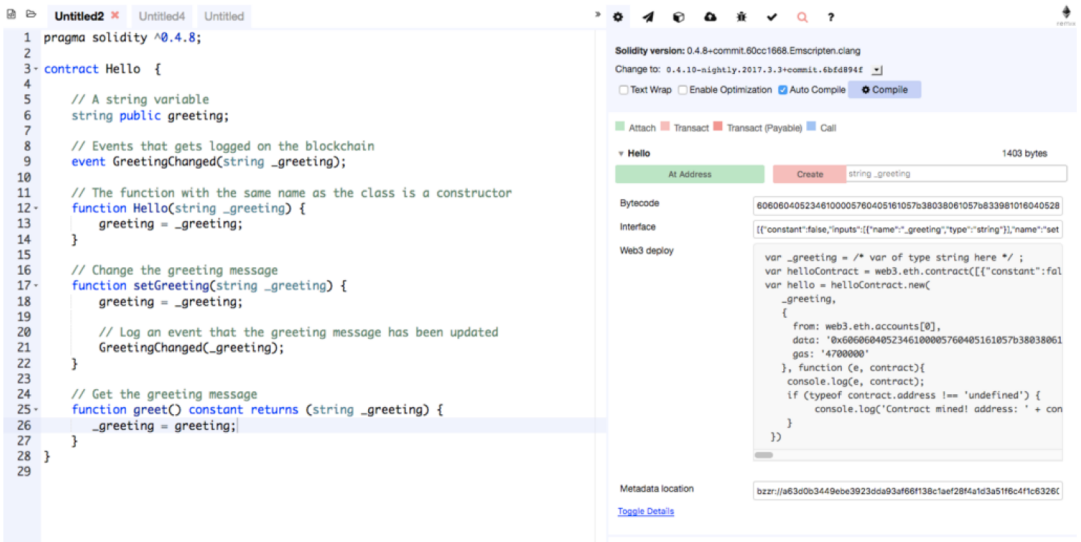
Why we're using Metamask?

1. In order to write on the Ethereum blockchain we need an Ethereum Node
2. Rather than installing the node on the computer, we're emulating the Ethereum Node in our web browser



Remix

- Development environment for the browser
- Easiest way to get started with development

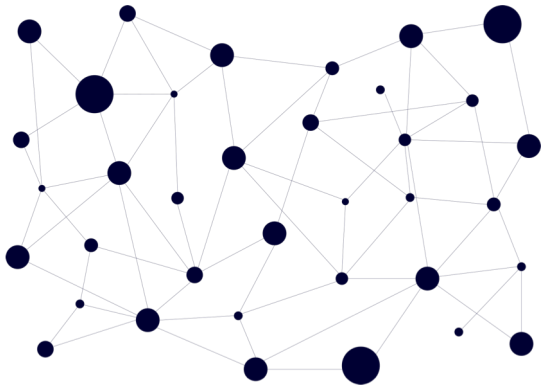


Ropsten

01001101
01010011
01010000

Why using a Test Network and not the Main network?

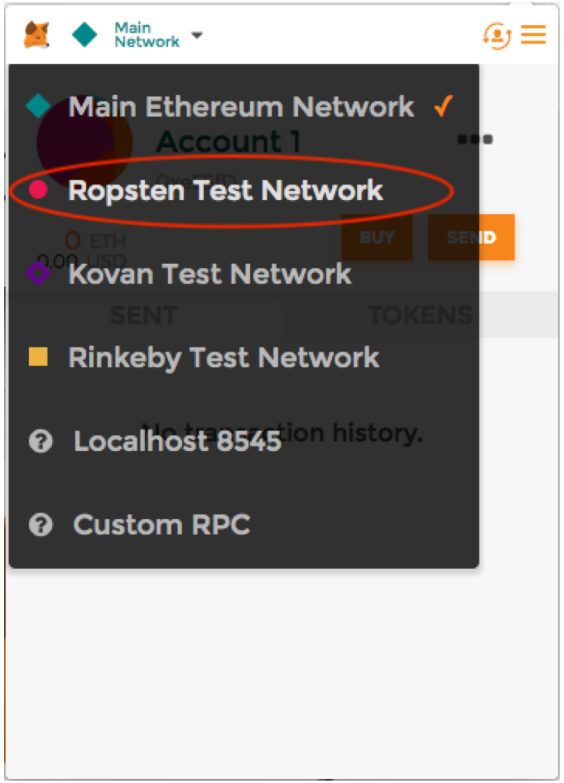
1. Deploy buggy code to the test network rather than the main network
2. Most up-to-date network
3. It's **free**



Setup

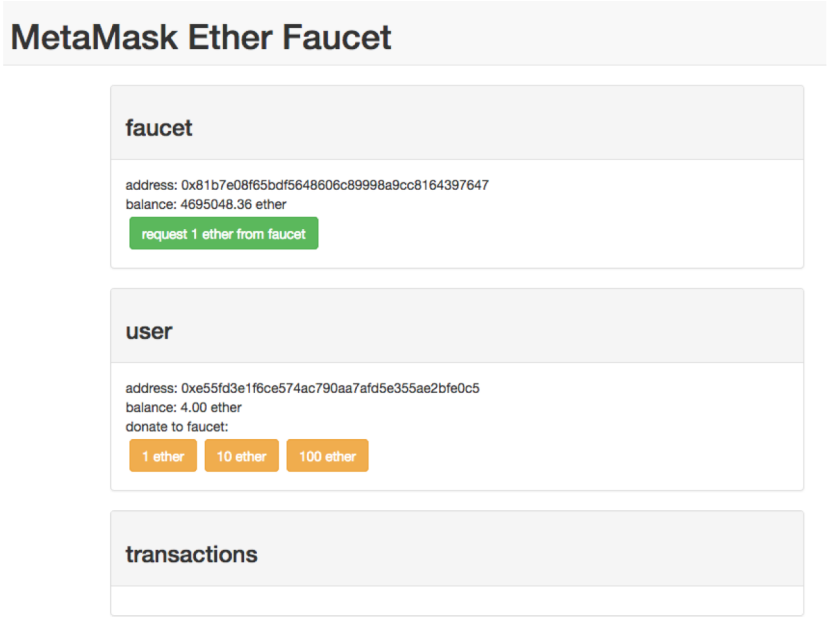
Setup – Metamask

1. Head to https://metamask.io/
2. Install the Browser Addon
3. Setup a password and safe the passphrase
4. Switch to the Ropsten Test Network



Setup Metamask

6. Head to https://faucet.metamask.io/
7. Request Ether for your Metamask wallet 💰💰💰



Etherscan

- Tool to see all transactions you've made on the Ethereum blockchain
- Visit https://ropsten.etherscan.io/

Setup - Remix

1. Head to https://remix.ethereum.org
2. Click on “Run” in the top left corner
3. Make sure you see your account address and are connected to Ropsten via Injected Web 3

Now you're ready to create your own Smart Contract!

Ethereum & Smart Contracts

Internet of Agreements

01001101
01010011
01010000

„Internet is to communication
what

Ethereum is to agreements“
(Gavin Wood, Co-Founder Ethereum)

Ethereum Smart Contracts

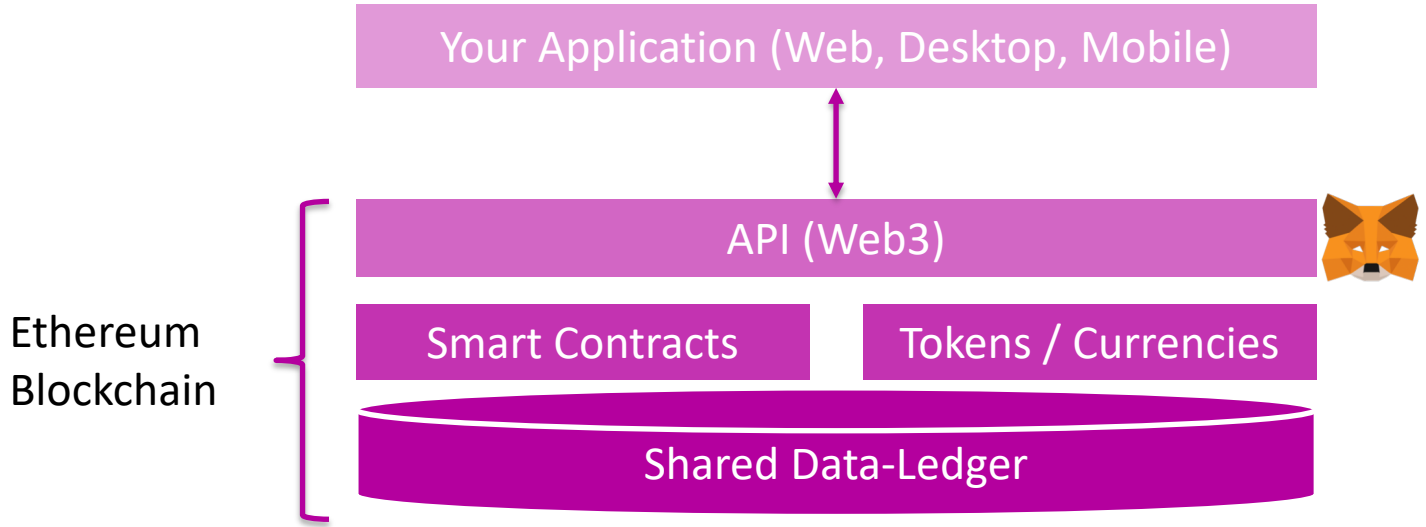
- Ethereum uses blockchain to implement **arbitrary social contracts** without central server
- Like Bitcoin, accounts with **balances**
- Unlike Bitcoin, accounts can be **contracts**, which hold:
  - **Code** to execute
  - **Storage** for structured data

Reasons to use blockchain Smart Contracts

01001101
01010011
01010000

- **Trust**
- **Autonomous**
- **Redundancy**
- **Savings**
- **Speed**
- **Transparency**
- **Precision**

Smart Contract: Application Architecture



Solidity

Solidity (programming language)

- Solidity: A contract oriented, high-level programming language for implementing smart contracts
- Influenced by
  - C++
  - Python
  - Javascript

Solidity: Key features

- Statically-typed
- Inheritance
- Libraries
- Complex user-defined types

Solidity: Basics

01001101
01010011
01010000

- Variables
- Functions
- Function modifiers
- Control Structures
- Message Object

Solidity: Structure of a contract

01001101
01010011
01010000

```
/* ------------------------------ Example contract -------------------------------*/

pragma solidity ^0.4.24
contract Example {
  //State variables

  //User defined types: Enums / Structs

  //Events

  //Custom function modifiers

  //Function definitions
}
```

Solidity: Variables

- State variables are **permantly** stored in **contract storage**
- Variable types:
  - Boolean: bool
  - Integers: int, uint, uint8, ...
  - Ethereum address: address (with members balance & transfer)
  - Arrays: uint[20] someArray

Solidity: Variables (cont.)

- Variable types:
  - Mappings: mapping (_keyType => _valueType)
  - String: string (dynamically-sized UTF-8 encoded string)

Solidity: Functions

- Function calls can happen internally and externally and have different levels of visibility
- Take set of input parameters and return set of output parameters

Solidity: Functions (Visibility)

- Visibility:
  - Public: Function can be called internally or via message
  - Private: Function can only be called from within contract (this does not include derived contracts)

Solidity: Functions (Types, Modifiers)

- Important types of functions:
  - View: Function does not modify state (of contract)
  - Payable: Functions that accept Ether sent with function call

Solidity: Message Object

- Special variable which always exists in the global namespace: **msg**
- Important members:
  - **msg.sender:** address that sent the message / function call (address)
  - **msg.value:** Amount of wei sent with message (uint)

„Hello – World“ Contract

Greeter Contract

Try it yourself!

Solidity: Further Functionalities

- Custom function modifiers
- Events
- Enum & Structs
- Control structures (if, else, while, do, for, break, continue)

Ethereum: Units of Ether

Name	Conversion Rate
wei	1 wei
kwei	10^3 wei
mwei	10^6 wei
gwei	10^9 wei
szabo	10^{12} wei
finney	10^{15} wei
ether	10^{18} wei

Contract Deployment: General Process

- Solidity compiler (solcjs) generates two types of files:
  - Application Binary Interface (ABI)
  - Bytecode
- Deployment: Bytecode is written to Ethereum Blockchain using a transaction

ABI & Bytecode

Ethereum Network: Gas (Transaction Fees)

- Every transaction is charged with a certain amount of gas (by transaction sender)
- Gas:
  - limits amount of work for the execution of a transaction
  - pays for the execution
- Gasprice: Price per gas in **ether**
- Total TX Fee = gasprice * gas

Voting Contract

Example Use Cases

Use Case I – District0x

Problem being solved: the value of companies often depends on the community, but the users cannot participate

How it works: kind of it's own little government with disctricts

smart contracts build an open source framework

provides basic functionalities of an online marketplace

getting rid of intermediates

Smart contract usage: users can only interact with district0x Network Tokens to protect the network

vote or pay with smart contracts

Use Case II – Grid+

01001101
01010011
01010000

Problem being solved: purchase electricity needs a middleman by now

How it works: a little iot device now purchases what you need

it buys and sells electricity on behalf of the user

it provides an accounting layer for the energy ecosystem

pushes market signals to the users to

Smart contract usage: the smart contracts are used to exchange of kWh to GIRD tokens to money

Use Case III – CryptoKitties

Problem being solved : there is never enough cat content!

How it works: collect, trade and breed cute kitties

it uses MetaMask to connect to the blockchain

the user can learn how to use smart contracts in a fun way

Smart contract usage: generation 0 kitties are distributed as smart contracts

Microsoft Student Partners

Microsoft

Microsoft Student Community

01001101
01010011
01010000

You want to keep in touch?

Get to know event dates earlier?

Win wildcards for exclusive MSP-events?

Join the Microsoft Student Community!

https://www.facebook.com/groups/MicrosoftStudentCommunity/