Ethereum Workshop

An Introduction to Tools, Solidity & Smart Contracts

Preparation

- Download and install:
  - Geth – an Ethereum client
    Installation instructions:
    https://github.com/ethereum/go-ethereum/wiki/Building-Ethereum
  - Mist – a DApp browser

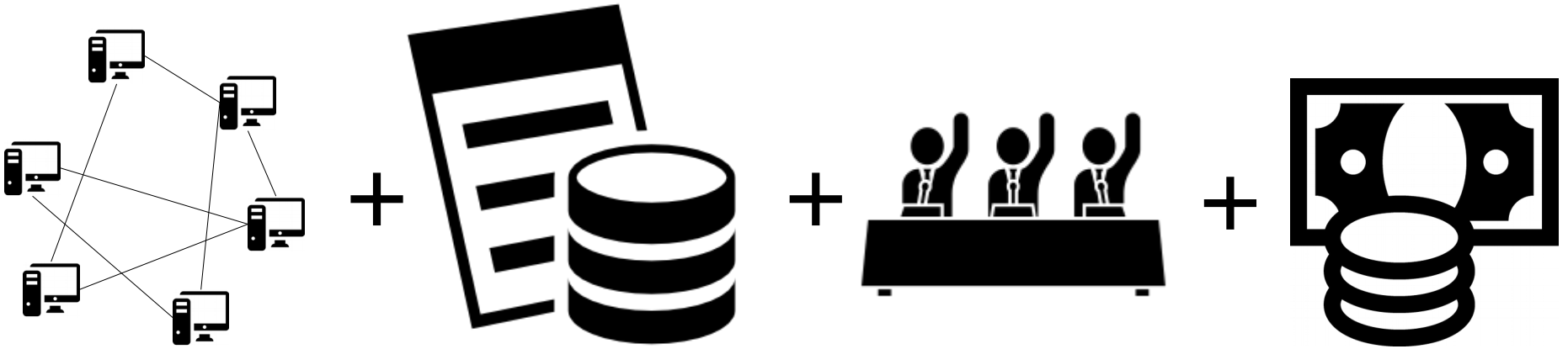
    Download the file starting with Mist-* (you need to scroll down a bit to find the list of files)
    https://github.com/ethereum/mist/releases
  - Download all files from:
    http://jonaspfannschmidt.com/eth_workshop/

Agenda

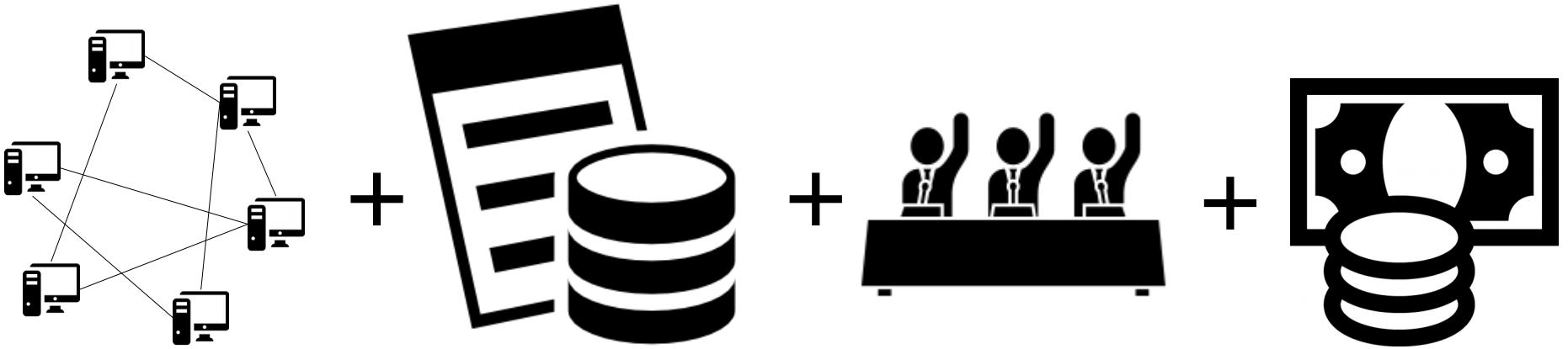
1) A brief introduction to Ethereum
2) Setting up a private blockchain
3) Interacting with the blockchain
4) Mist
5) Solidity & Smart Contracts
6) Remix IDE

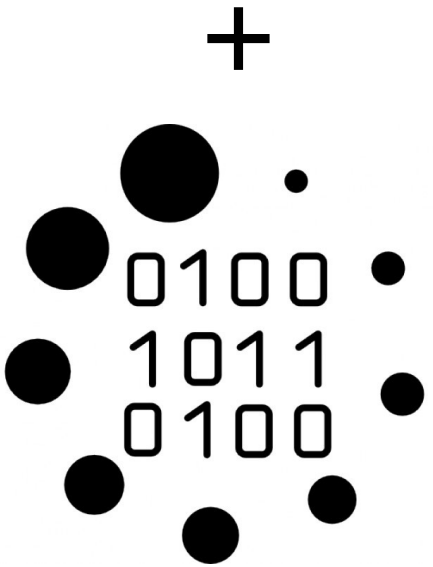
1. A brief introduction to Ethereum

Blockchain Introduction



Etherum

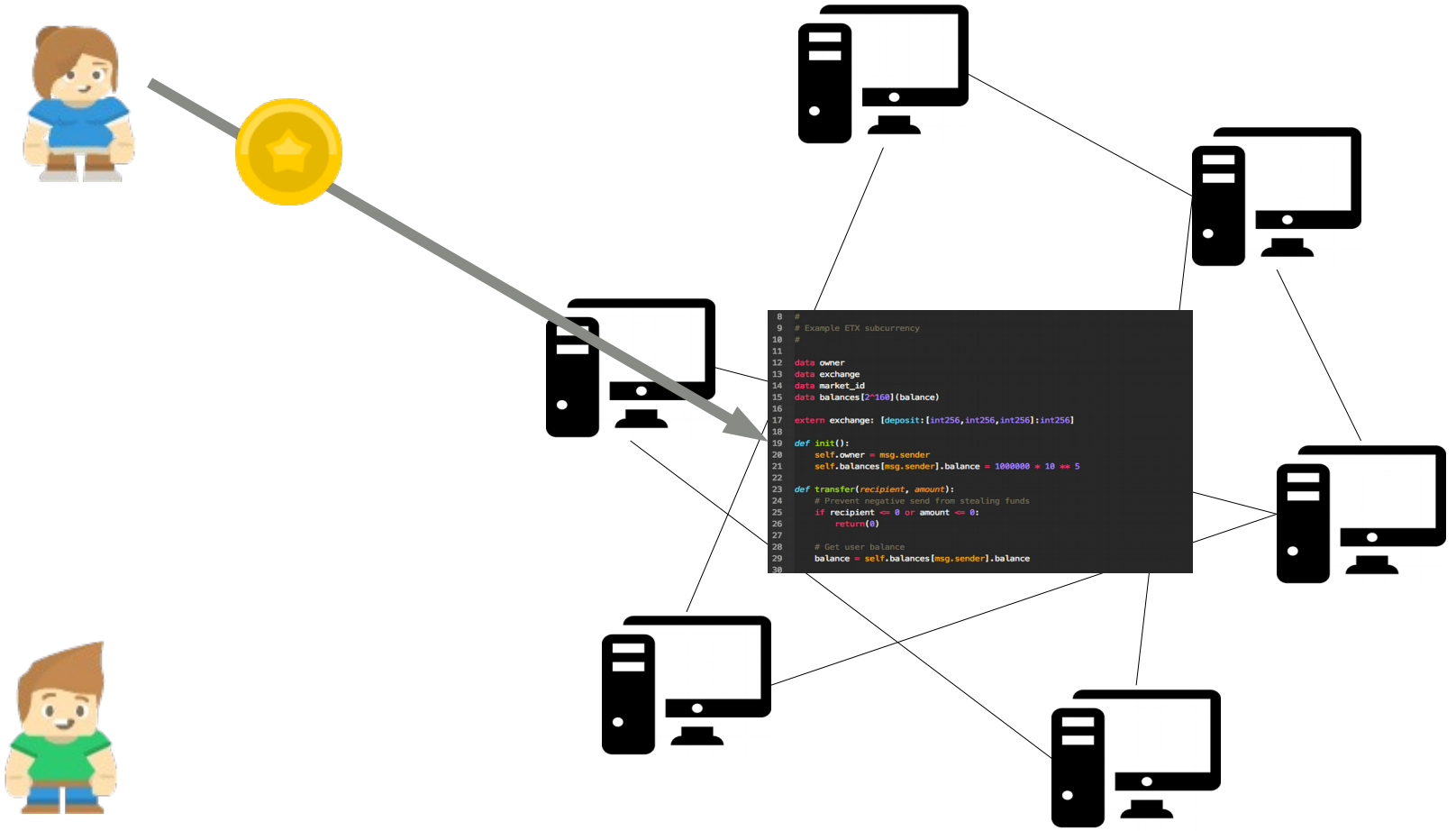




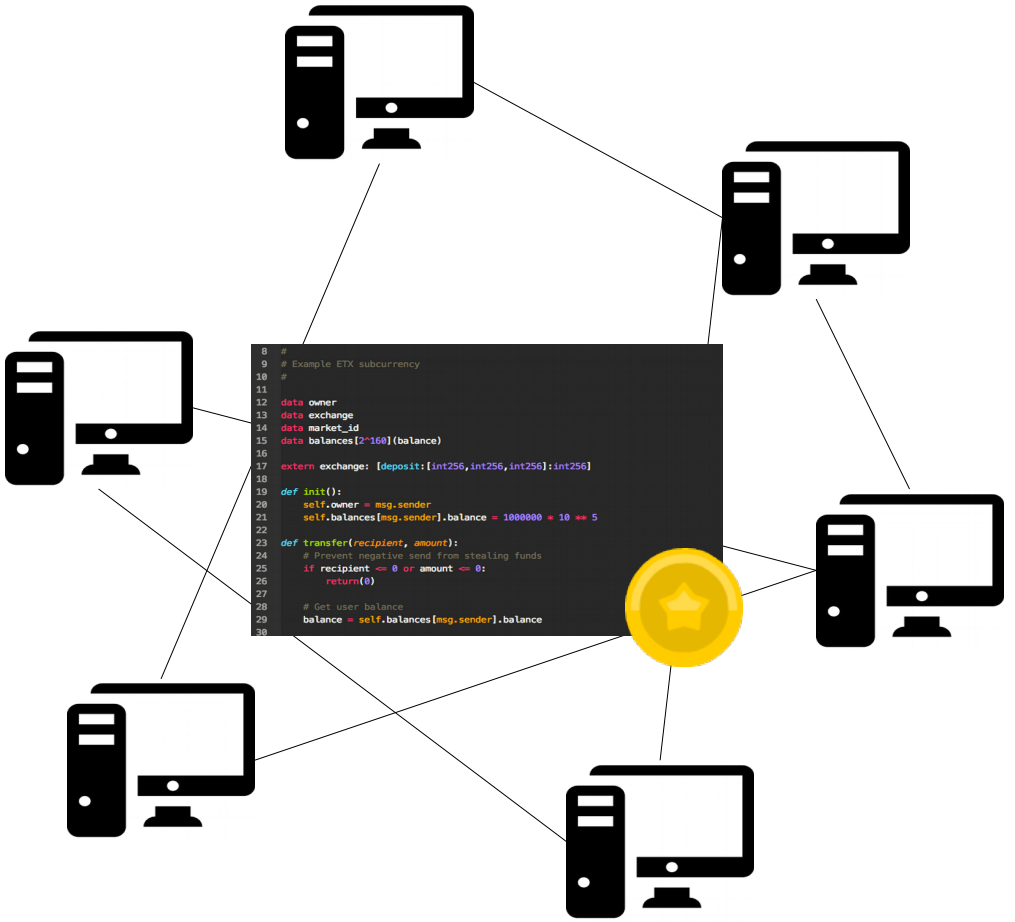
Why?

```
#
# Example ETX subcurrency
#

data owner
data exchange
data market_id
data balances[2^160](balance)

extern exchange: [deposit:[int256,int256,int256]:int256]

def init():
    self.owner = msg.sender
    self.balances[msg.sender].balance = 1000000 * 10 ** 5

def transfer(recipient, amount):
    # Prevent negative send from stealing funds
    if recipient <= 0 or amount <= 0:
        return(0)

    # Get user balance
    balance = self.balances[msg.sender].balance

```

Why?


#
Example ETX subcurrency
#
data owner
data exchange
data market_id
data balances[2^160](balance)
extern exchange: [deposit:[int256,int256,int256]:int256]
def init():
self.owner = msg.sender
self.balances[msg.sender].balance = 1000000 * 10 ** 5
def transfer(recipient, amount):
# Prevent negative send from stealing funds
if recipient <= 0 or amount <= 0:
return(0)
# Get user balance
balance = self.balances[msg.sender].balance

Why?


#
Example ETX subcurrency
#
data owner
data exchange
data market_id
data balances[2^160](balance)
extern exchange: [deposit:[int256,int256,int256]:int256]
def init():
self.owner = msg.sender
self.balances[msg.sender].balance = 1000000 * 10 ** 5
def transfer(recipient, amount):
# Prevent negative send from stealing funds
if recipient <= 0 or amount <= 0:
return(0)
# Get user balance
balance = self.balances[msg.sender].balance

2. Setting up a private blockchain

Accounts

```
> geth --datadir ~/.ethereum/workshop
  account new
```

- `geth` – **G**o **Eth**ereum client
- `--datadir <DIRECTORY>` – Store all data (incl. the blockchain in DIRECTOY)
- `account new` – Create a new account

Windows!

- On Windows machines, replace:

  `~/.ethereum/workshop`

  with:

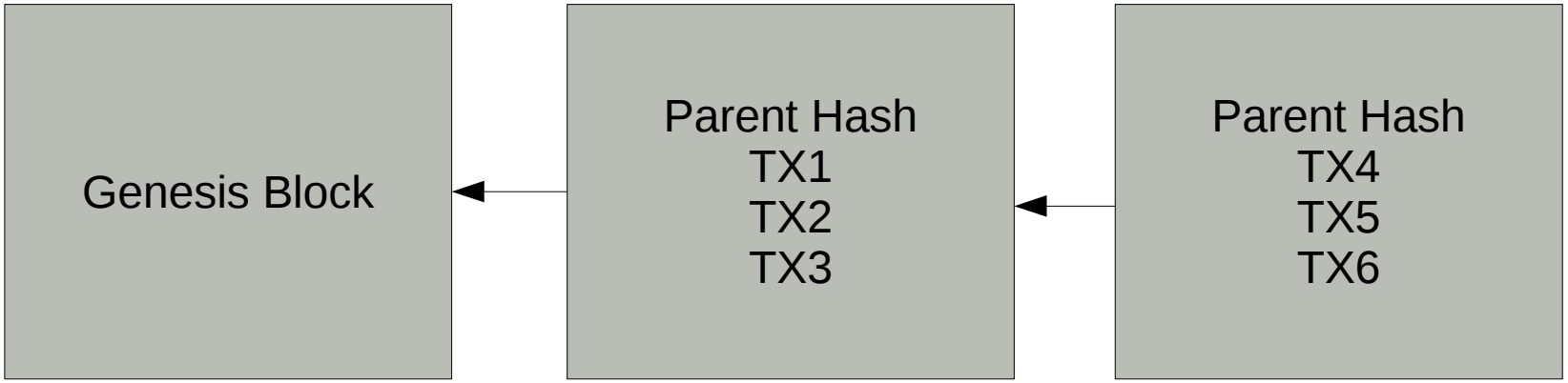
- `C:\Users\[USERNAME]\workshop`

Accounts

```
> geth --datadir ~/.ethereum/workshop
  account list
```

- `account list` – List all existing accounts

Blockchain Data Structure



The genesis block

```
{
    "config": {
        "chainId": 42,
        "homesteadBlock": 0,
        "eip155Block": 0,
        "eip158Block": 0
    },
    "nonce": "0x0000000000000042",
    "timestamp": "0x00",
    "parentHash": "0x000000000[...]00000000000",
    "extraData": "0x00",
    "gasLimit": "0x4c4b40",
    "difficulty": "0x6666",
    "mixhash": "0x000000000000[...]00000000000",
    "coinbase": "0x00000000000[...]00000000000"
}
```

https://ethereum.stackexchange.com/questions/2376/what-does-each-genesis-json-parameter-mean
https://ethereum.stackexchange.com/questions/15682/the-meaning-specification-of-config-in-genesis-json/15687#15687
https://ethereum.stackexchange.com/questions/5833/why-do-we-need-both-nonce-and-mixhash-values-in-a-block

Initialize the blockchain

```
> geth --datadir ~/.ethereum/workshop
  init genesis.json
```

- `init <GENESIS FILE>` – Initialize a new blockchain from a genesis file

Start a miner

```
> geth --datadir ~/.ethereum/workshop
  --networkid 42 --mine --minerthreads
  1
```

- `--networkid <NUMBER>` – A unique number for this blockchain network
- `--mine` – Make this blockchain node a miner
- `--minerthreads <NUMBER>` – How many threads (which indirectly means CPUs) are used for mining

Predefined network ids

1) Frontier
2) Morden (disused)
3) Ropsten
4) Rinkeby

Plan B if something doesn‘t work

```
> geth --dev
```

- `--dev` – Developer mode: pre-configured private network – Cannot connect to other nodes

Start a console

```
> geth attach
ipc:///home/jonas/.ethereum/workshop/g
eth.ipc
```

- `attach <PATH>` – Attach a console to a running geth instance using IPC

Windows!

- On Windows machines run just:
  `> geth attach`

2. Interacting with the blockchain

admin API

- `admin.nodeInfo` – Gives us the enode id and a bunch of useful information
- `admin.peers` – Lists all connected nodes our node knows
- `admin.addPeer("enode://fc[...]03")` – Manually add another node

https://github.com/ethereum/go-ethereum/wiki/JavaScript-Console#management-api-reference

personal API

- `personal.listAccounts` – List of all (local) accounts
- `personal.unlockAccount("0xc73[...]5b")` – Lists all connected nodes our node knows

https://github.com/ethereum/go-ethereum/wiki/JavaScript-Console#management-api-reference

web3js

- `web3.eth.getBalance("0xc[...]5b")` – Get balance of account. This works for all accounts.

  https://github.com/ethereum/wiki/wiki/JavaScript-API#web3js-api-reference

Dealing with numbers

- `number.toString(10)` – Converts a bignumber to a human-readable string
  https://github.com/ethereum/wiki/wiki/JavaScript-API#a-note-on-big-numbers-in-web3js
  https://github.com/MikeMcl/bignumber.js/
- `web3.fromWei(number, "ether")` – Converts from wei to ether

  https://github.com/ethereum/wiki/wiki/JavaScript-API#web3fromwei
  https://converter.murkin.me/
- `web3.toWei(number, "ether")` – From ether to wei

web3js

- web3.eth.sendTransaction(
  {"from": "0xc73e[...]2cfbc025b",
  "to": "0x00[...]00",
  "value": 111111
  })
  – Send wei from an address to another address

3. Mist

Starting Mist

```
> mist --rpc
/home/jonas/.ethereum/workshop/geth.ipc
```

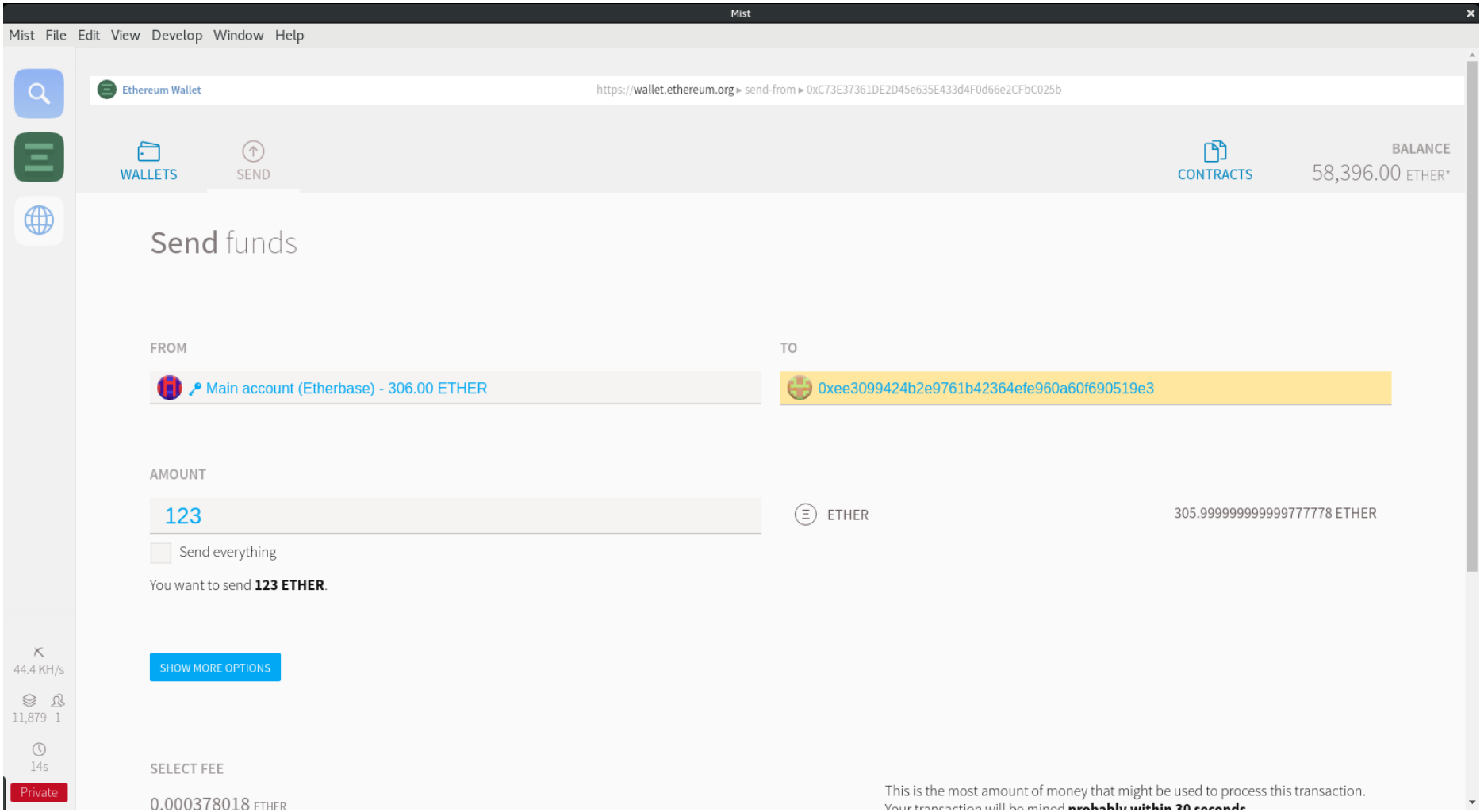
- `--rpc` – Path to node IPC socket file OR HTTP RPC hostport

Windows!

- On Windows machines run:

```
> "C:\Program Files\Mist\Mist.exe"
--rpc \\.\pipe\geth.ipc
```

Mist


Mist
Mist File Edit View Develop Window Help
Ethereum Wallet
https://wallet.ethereum.org ▸ send-from ▸ 0xC73E37361DE2D45e635E433d4F0d66e2CFbC025b
WALLETS
SEND
CONTRACTS
BALANCE
58,396.00 ETHER*
Send funds
FROM
Main account (Etherbase) - 306.00 ETHER
TO
0xee3099424b2e9761b42364efe960a60f690519e3
AMOUNT
123
ETHER
305.999999999999777778 ETHER
Send everything
You want to send 123 ETHER.
SHOW MORE OPTIONS
44.4 KH/s
11,879 1
14s
Private
SELECT FEE
0.000378018 ETHER
This is the most amount of money that might be used to process this transaction.

4. Solidity & Smart Contracts

Smart Contracts

```
pragma solidity ^0.4.15;

contract MyCoin {
    mapping (address => uint) balances;

    function MyCoin() {
        balances[tx.origin] = 10000;
    }

    function sendCoin(address receiver, uint amount) returns(bool sufficient) {
        if (balances[msg.sender] < amount) return false;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        return true;
    }

    function getBalance(address addr) returns(uint) {
        return balances[addr];
    }
}
```

Smart Contracts

```
pragma solidity ^0.4.15;

contract MyCoin {
    mapping (address => uint) balances;

    function MyCoin() {
        balances[tx.origin] = 10000;
    }

    function sendCoin(address receiver, uint amount) returns(bool sufficient) {
        if (balances[msg.sender] < amount) return false;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        return true;
    }

    function getBalance(address addr) returns(uint) {
        return balances[addr];
    }
}
```

Smart Contracts

```
pragma solidity ^0.4.15;

contract MyCoin {
    mapping (address => uint) balances;

    function MyCoin() {
        balances[tx.origin] = 10000;
    }

    function sendCoin(address receiver, uint amount) returns(bool sufficient) {
        if (balances[msg.sender] < amount) return false;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        return true;
    }

    function getBalance(address addr) returns(uint) {
        return balances[addr];
    }
}
```

Smart Contracts

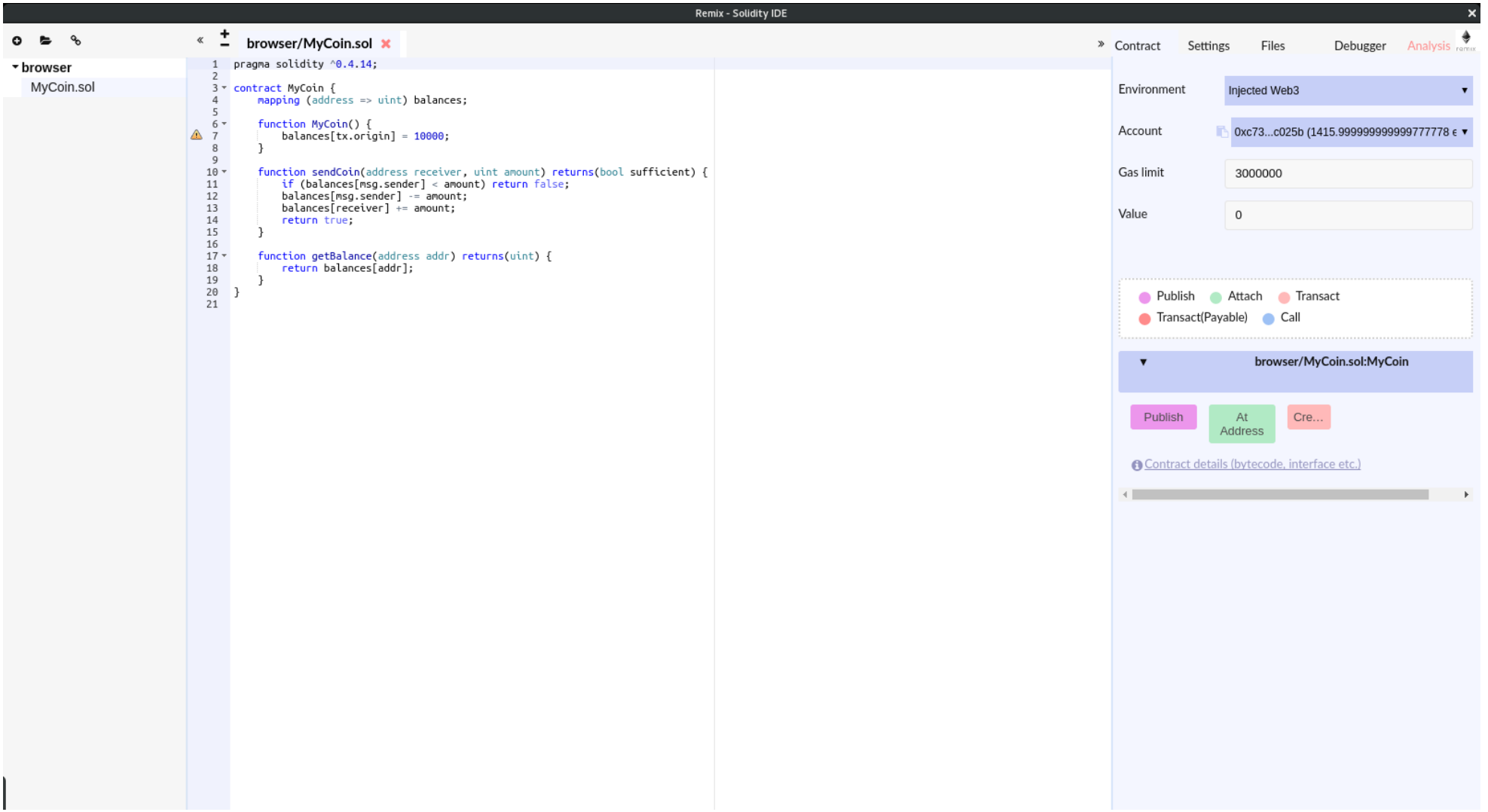
```
pragma solidity ^0.4.15;

contract MyCoin {
    mapping (address => uint) balances;

    function MyCoin() {
        balances[tx.origin] = 10000;
    }

    function sendCoin(address receiver, uint amount) returns(bool sufficient) {
        if (balances[msg.sender] < amount) return false;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        return true;
    }

    function getBalance(address addr) returns(uint) {
        return balances[addr];
    }
}
```

Smart Contracts

```
pragma solidity ^0.4.15;

contract MyCoin {
    mapping (address => uint) balances;

    function MyCoin() {
        balances[tx.origin] = 10000;
    }

    function sendCoin(address receiver, uint amount) returns(bool sufficient) {
        if (balances[msg.sender] < amount) return false;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        return true;
    }

    function getBalance(address addr) returns(uint) {
        return balances[addr];
    }
}
```

5. Remix IDE

Remix


Remix - Solidity IDE
browser
MyCoin.sol
browser/MyCoin.sol
pragma solidity ^0.4.14;
contract MyCoin {
mapping (address => uint) balances;
function MyCoin() {
balances[tx.origin] = 10000;
}
function sendCoin(address receiver, uint amount) returns(bool sufficient) {
if (balances[msg.sender] < amount) return false;
balances[msg.sender] -= amount;
balances[receiver] += amount;
return true;
}
function getBalance(address addr) returns(uint) {
return balances[addr];
}
}
Contract
Settings
Files
Debugger
Analysis
Environment
Injected Web3
Account
0xc73...c025b (1415.99999999999777778
Gas limit
3000000
Value
0
Publish
Attach
Transact
Transact(Payable)
Call
browser/MyCoin.sol:MyCoin
Publish
At Address
Cre...
Contract details (bytecode, interface etc.)

Questions?