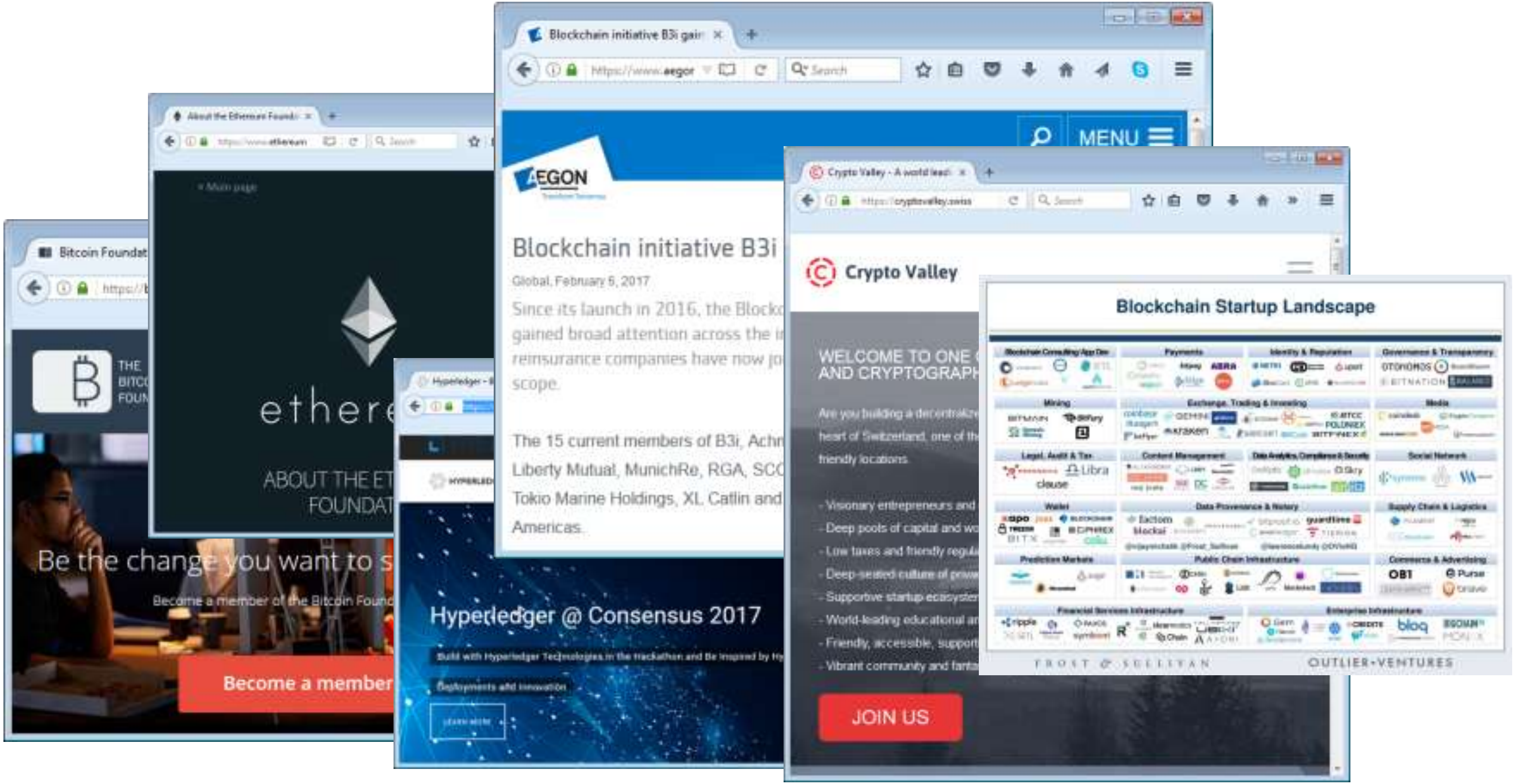
Blockchain, Ethereum and Business Applications

@ZimMatthias Matthias Zimmermann
BSI Business Systems Integration AG



Bitcoin Foundat
THE BITC FOUN
Be the change you want to s
Become a member of the Bitcoin Found
Become a member
About the Ethereum Found
‹ Main page
ethere
ABOUT THE ET FOUNDAT
Hyperledger
HYPERLED
Hyperledger @ Consensus 2017
Build with Hyperledger Technologies in the Hackathon and Be Inspired by Hy
Deployments and Innovation
LEARN MORE
Blockchain initiative B3i gair
https://www.aegon
MENU
AEGON
Crypto Valley - A world lead
https://cryptovalley.swiss
Crypto Valley
JOIN US
Blockchain Startup Landscape
Payments
Mining
Exchange, Trading & Investing
Media
Legal, Audit & Tax
Content Management
Social Network
Wallet
Data Provenance & Notary
Supply Chain & Logistics
Prediction Markets
Public Chain Infrastructure
Commerce & Advertising
Financial Services Infrastructure
Enterprise Infrastructure
FROST & SULLIVAN
OUTLIER VENTURES

Blockchain initiative B3i

Global, February 6, 2017

Since its launch in 2016, the Blocke gained broad attention across the in reinsurance companies have now jo scope.

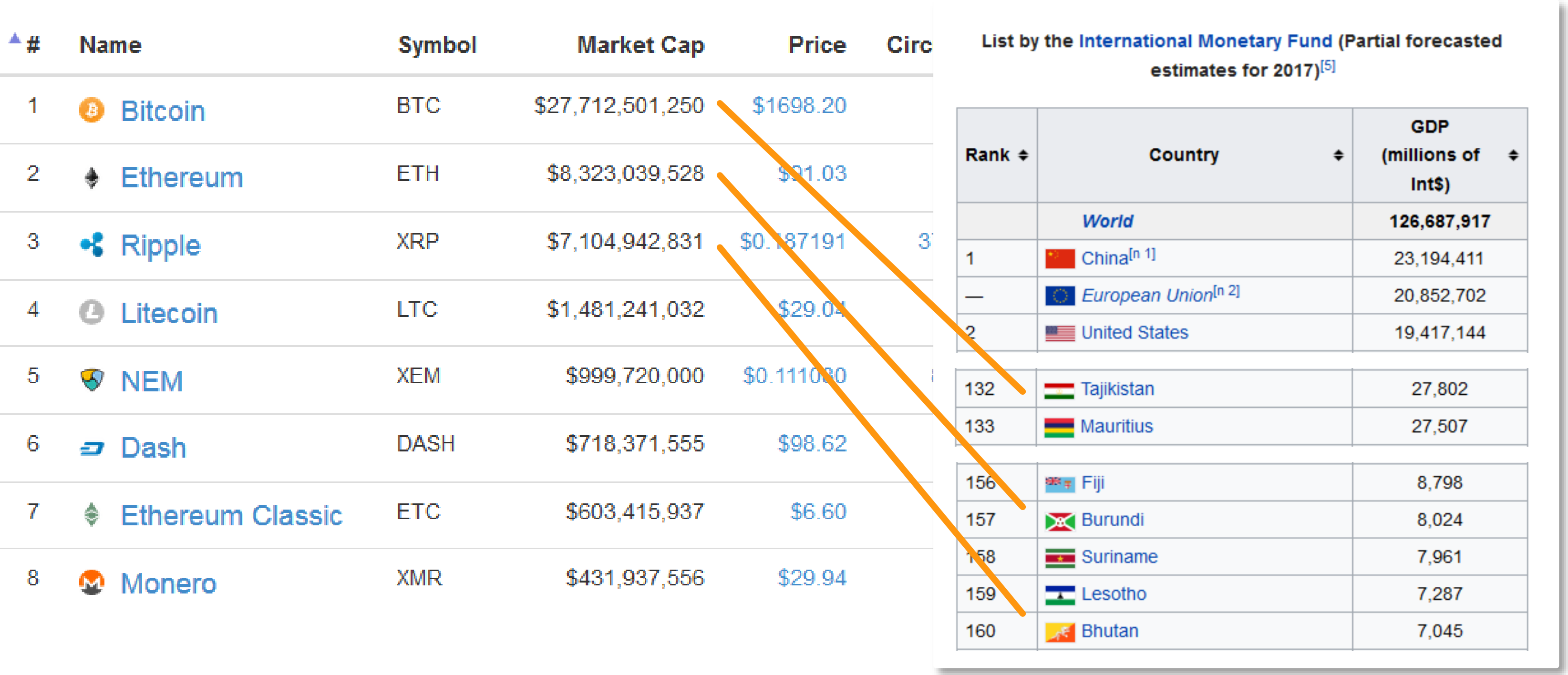
The 15 current members of B3i, Achn Liberty Mutual, MunichRe, RGA, SCC Tokio Marine Holdings, XL Catlin and Americas.

WELCOME TO ONE (AND CRYPTOGRAPH

Are you building a decentralize heart of Switzerland, one of th friendly locations.

- Visionary entrepreneurs and
- Deep pools of capital and wo
- Low taxes and friendly regula
- Deep-seated culture of prive
- Supportive startup ecosysten
- World-leading educational ar
- Friendly, accessible, support
- Vibrant community and fanta

Crypto Currencies Market



#	Name	Symbol	Market Cap	Price	Circ
1	Bitcoin	BTC	$27,712,501,250	$1698.20	
2	Ethereum	ETH	$8,323,039,528	$91.03	
3	Ripple	XRP	$7,104,942,831	$0.187191	3
4	Litecoin	LTC	$1,481,241,032	$29.04	
5	NEM	XEM	$999,720,000	$0.111080	
6	Dash	DASH	$718,371,555	$98.62	
7	Ethereum Classic	ETC	$603,415,937	$6.60	
8	Monero	XMR	$431,937,556	$29.94	

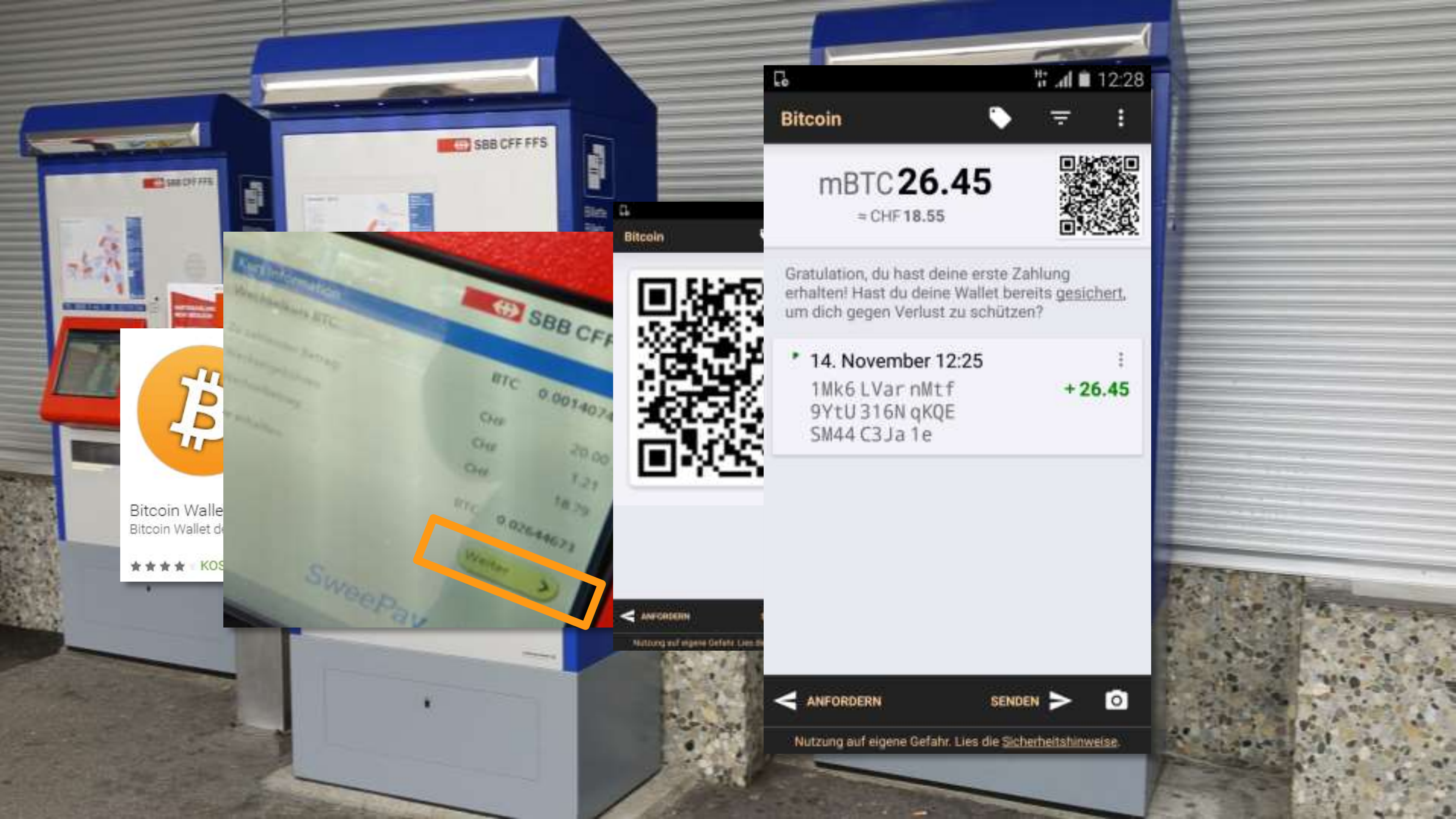
List by the International Monetary Fund (Partial forecasted estimates for 2017)[5]

Rank	Country	GDP (millions of Int$)
	World	**126,687,917**
1	China[n 1]	23,194,411
—	*European Union*[n 2]	20,852,702
2	United States	19,417,144
132	Tajikistan	27,802
133	Mauritius	27,507
156	Fiji	8,798
157	Burundi	8,024
158	Suriname	7,961
159	Lesotho	7,287
160	Bhutan	7,045

Bitcoin

First decentralized digital currency

- By «Satoshi Nakamoto»
- White Paper 2008
- Open Source Software 2009



mBTC 26.45

≈ CHF 18.55

Gratulation, du hast deine erste Zahlung erhalten! Hast du deine Wallet bereits gesichert, um dich gegen Verlust zu schützen?

14. November 12:25
1Mk6 LVar nMtf
9YtU 316N qKQE
SM44 C3Ja 1e
+ 26.45

Nutzung auf eigene Gefahr. Lies die Sicherheitshinweise.

Coffee at Bob‘s

Address	**1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA**
Amount [₿]	**0.015**
Name Recipient	**Bob's Café**

Source: «Mastering Bitcoin», *Andreas M. Antonopoulos*

What is a Bitcoin Address?

Addresses corresponds to **«Accounts»**

Encoded Numbers

➔ Example: `1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA`

➔ Derived from a **private Key**

➔ Private key is 256-Bit random number

Getting, using and loosing Accounts

➔ Create your account with «Coin, Paper and Pencil»

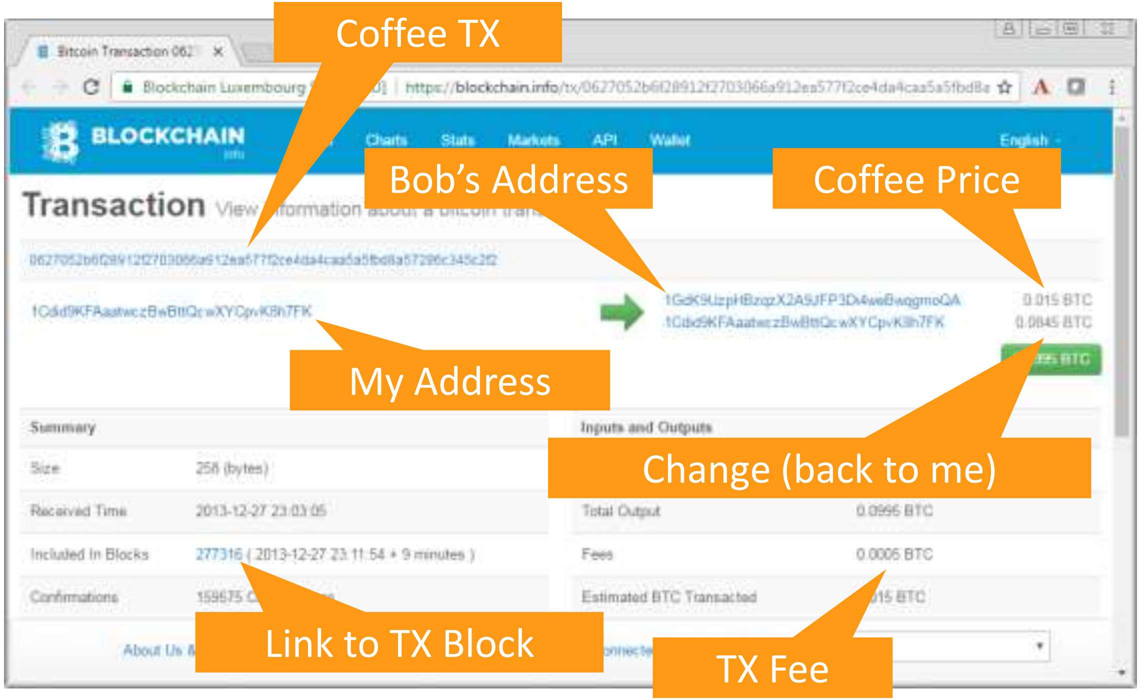
➔ No ID required, no showing up at local branch, …

➔ To send Bitcoins you need your private Key

➔ **You loose your private key → you loose your money**

The Coffee Transaction

- ➔ Bitcoin network confirms coffee TX after ~ 10 min.
- ➔ **TX Elements**



Bitcoin Transaction 062

Blockchain Luxembourg https://blockchain.info/tx/0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8e

Transaction View information about a bitcoin tran

0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8a57286c345c2f2

1Cdid9KFAaatwczBwBttQcwXYCpvK8h7FK ➔ 1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA 0.015 BTC
1Cdid9KFAaatwczBwBttQcwXYCpvK8h7FK 0.0845 BTC

995 BTC

Summary	
Size	258 (bytes)
Received Time	2013-12-27 23:03:05
Included In Blocks	277316 (2013-12-27 23:11:54 + 9 minutes)
Confirmations	159575 C

Inputs and Outputs	
Total Output	0.0995 BTC
Fees	0.0005 BTC
Estimated BTC Transacted	015 BTC

About Us &

Coffee TX

Bob’s Address

Coffee Price

My Address

Change (back to me)

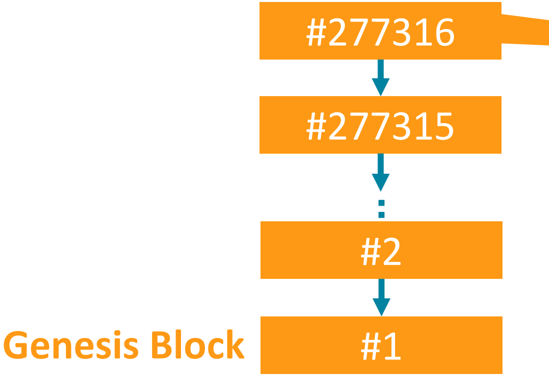
Link to TX Block

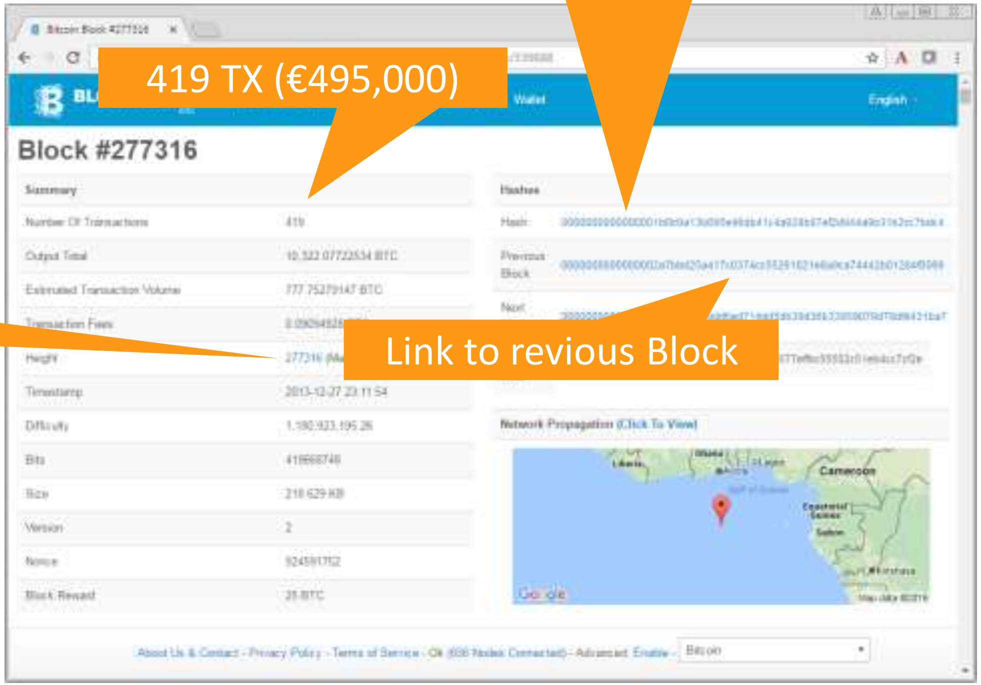
TX Fee

Screenshot: blockchain.info

TX Blocks and the Blockchain

- ➔ **Block** #277316 includes coffee TX
- ➔ Block Elements





Block #277316

Summary	
Number Of Transactions	419
Output Total	10,322.07722534 BTC
Estimated Transaction Volume	777.75279147 BTC
Transaction Fees	0.0909492[illegible]
Height	277316 [illegible]
Timestamp	2013-12-27 23:11:54
Difficulty	1,180,923,195.26
Bits	419668748
Size	218.629 KB
Version	2
Nonce	924591752
Block Reward	25 BTC

Hashes	
Hash	0000000000000001b6b9a13b095e96db41c4a928b97ef2d944a9b31b2cc7bdc4
Previous Block	0000000000000002a7bbd25a417c0374cc55261021e8a9ca74442b01284f0569
Next Block	[illegible]
[illegible]	[illegible]55502c51eb4cc7cf2e

Network Propagation (Click To View)

About Us & Contact - Privacy Policy - Terms of Service - Ok (838 Nodes Connected) - Advanced: Enable - Bitcoin

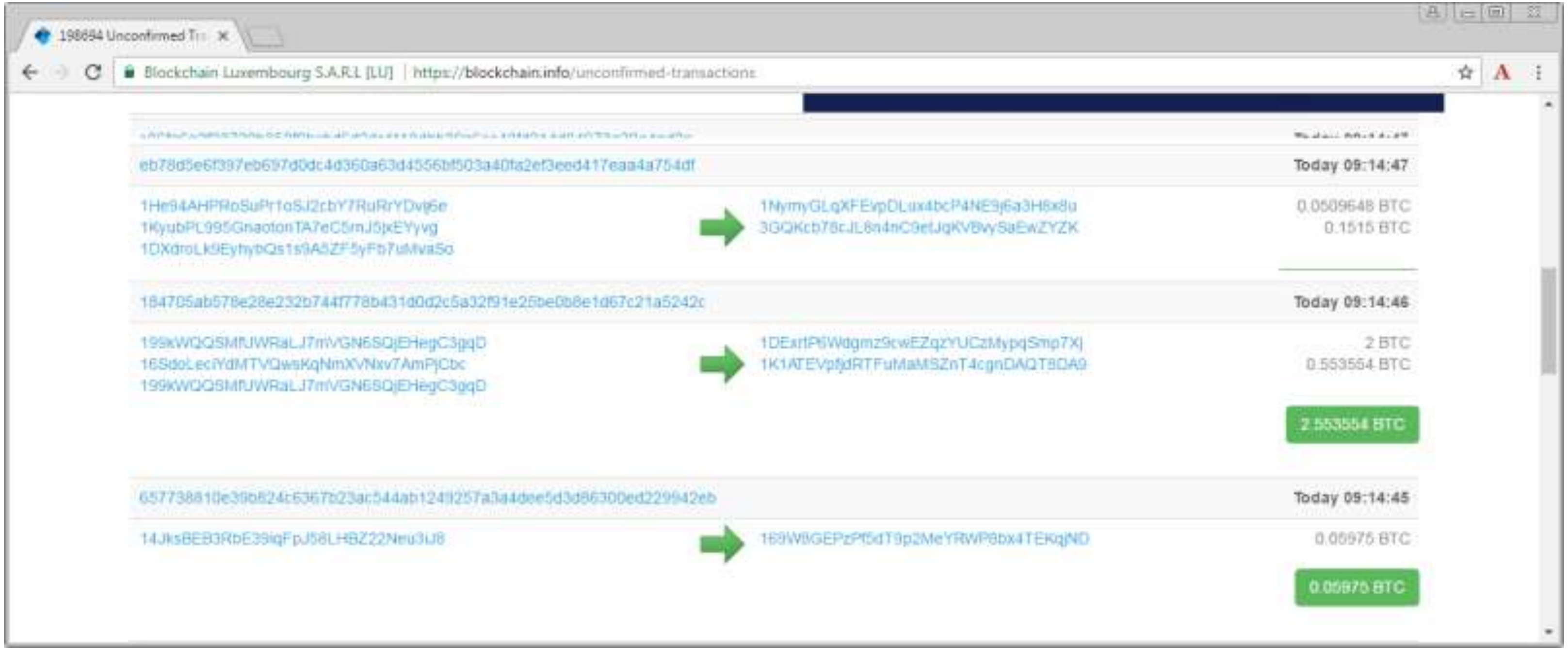
419 TX (€495,000)

Block Hash – do you see the leading zeros?

Link to revious Block

Screenshot: blockchain.info

From Transactions to Blocks (Mining)


198694 Unconfirmed Tr
Blockchain Luxembourg S.A.R.L [LU]
https://blockchain.info/unconfirmed-transactions

eb78d5e6f397eb697d0dc4d360a63d4556bf503a40fa2ef3eed417eaa4a754df Today 09:14:47

1He94AHPRoSuPr1oSJ2cbY7RuRrYDvj6e
1KyubPL995GnaotonTA7eC5mJ5jxEYyvg
1DXdroLk9EyhybQs1s9A5ZF5yFb7uMva5o
→
1NymyGLqXFEvpDLux4bcP4NE9j6a3H6x8u 0.0509648 BTC
3GQKcb78cJL8n4nC9etJqKVBvySaEwZYZK 0.1515 BTC

184705ab578e28e232b744f778b431d0d2c5a32f91e25be0b8e1d67c21a5242c Today 09:14:46

199kWQQSMfUWRaLJ7mVGN6SQjEHegC3gqD
16SdoLeciYdMTVQwsKqNmXVNxv7AmPjCbc
199kWQQSMfUWRaLJ7mVGN6SQjEHegC3gqD
→
1DExrtP6Wdgmz9cwEZqzYUCzMypqSmp7Xj 2 BTC
1K1ATEVpfjdRTFuMaMSZnT4cgnDAQT8DA9 0.553554 BTC

2.553554 BTC

657738810e39b824c6367b23ac544ab1249257a3a4dee5d3d86300ed229942eb Today 09:14:45

14JksBEB3RbE39iqFpJ58LHBZ22Neu3iJ8
→
169W8GEPzPf5dT9p2MeYRWP8bx4TEKqjND 0.05975 BTC

0.05975 BTC

From Transactions to Blocks (Mining)

1. New TX are propagatet through Bitcoin **peer-to-peer network**
2. Bitcoin client verifies new TX and adds it to local **«mempool»**
3. Client starts to **«mine»** transactions:
    - Assemble TX from mempool to **block candidate**
    - Starts to solve the block candidate's **crypto challenge**
    - Computes **MANY** hashes to solve the crypto challenge
    - Client solving the challenge first, gets **block reward** and all **TX fees**

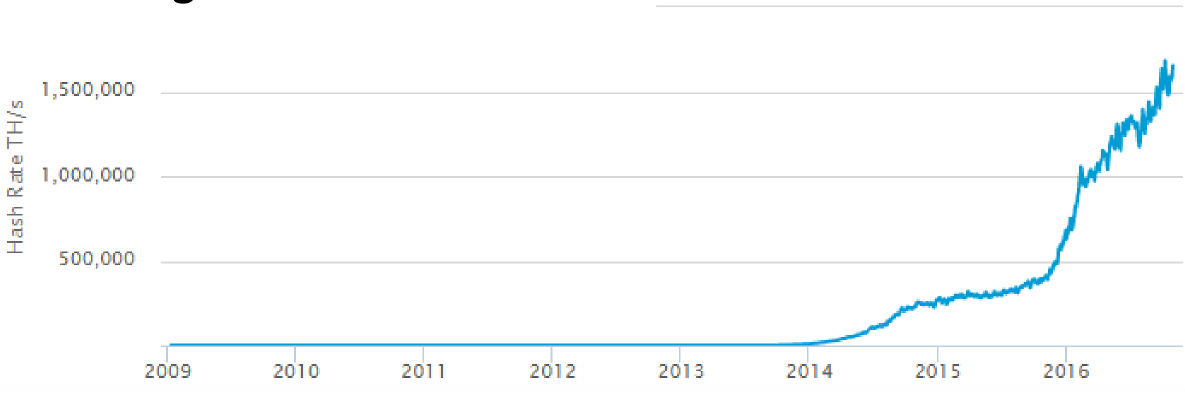
3. Winning client sends the new block to its peers
4. Arrival of new block triggers the next challenge

Creation of new Bitcoins

Bitcoin Mining Today

- ➔ **Mining-pools**: Include many ASIC computers (PC way too slow)
- ➔ **AntMiner**: 10,000x faster than PC, burns 10x more electricity
- ➔ **Energy Costs**: # of hashes per KWh is central criteria + cooling(!)

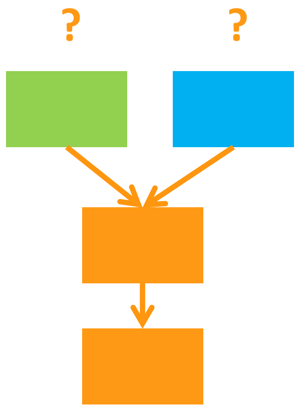
Hashing Power over the Years



1.5×10^{18}

IGraph: blockchain.info

Distributed Consensus Mechanism
Preventing Forks



The Challenge

- ➔ Mining clients build block candidates independently
- ➔ **Several new blocks** might be found **at the «same» time**
- ➔ Clients may receive new blocks that are inconsistent
- ➔ The local copy of the blockchain may have forks

The Solution

- ➔ The «true» blockchain is defined by the **highest cumulative PoW** (difficulty)
- ➔ By selecting the greatest-difficulty chain, **eventual consensus** is achieved
- ➔ Miner **majority vote** defines the true chain
- ➔ Miners «vote» for the true chain by deciding which block/fork to extend

«Unhappy with some TX?»
The 51% Attack

Attacking Bitcoin (any PoW Blockchain)

1. Install more mining capacity than the rest of the world (>= 51%)
2. Censor/suppress unwanted TX
3. Mine a secret branch containing acceptable TX
4. Continue to mine until PoW of secret branch exceeds official branch
5. Broadcast secret branch to Bitcoin network
6. All Bitcoin clients will switch to this new branch

Ok, this is kind of hard – but:

➔ Miners earn ~1.2bn/year

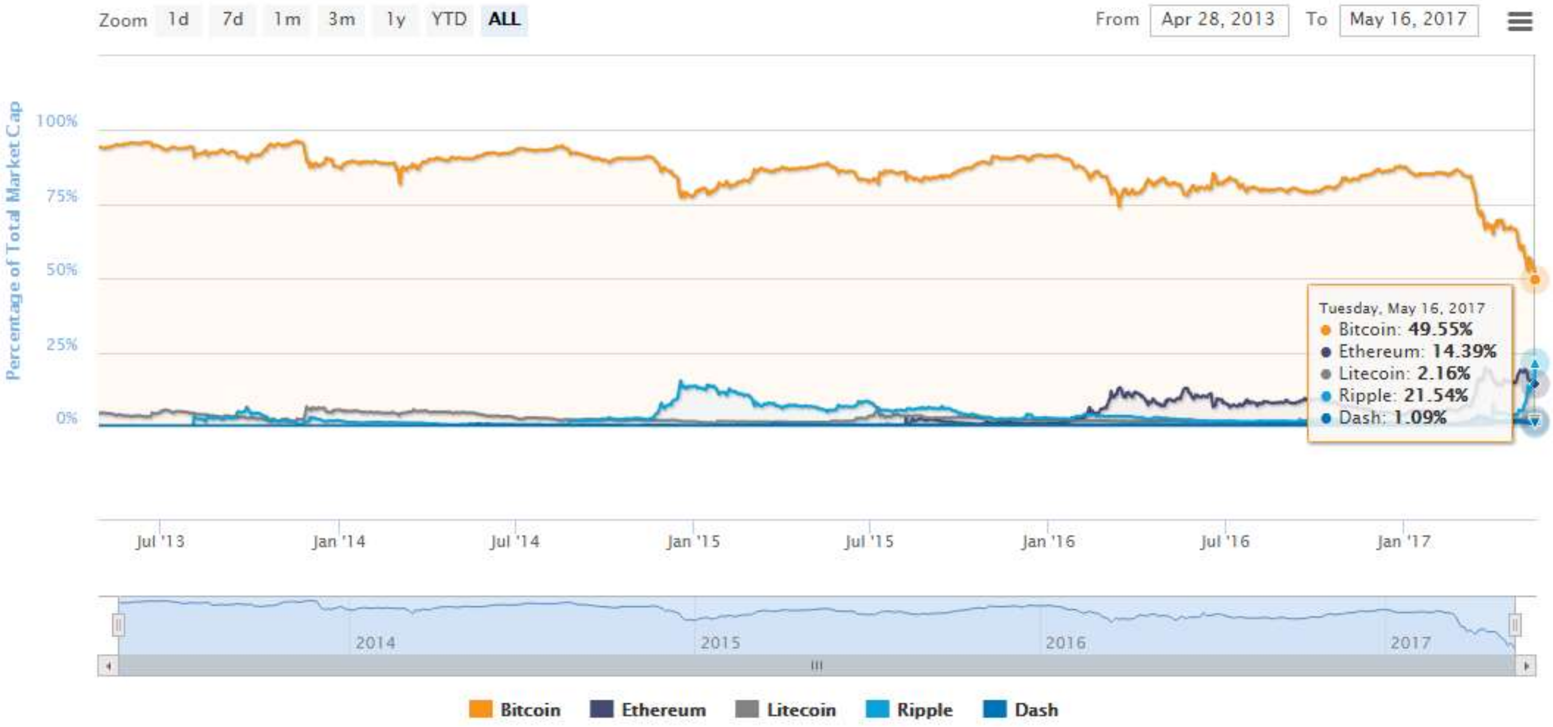
➔ HW cost to match Bitcoin mining capacity: ~ $400m (Antminer 9s)

Bitcoin Challenges

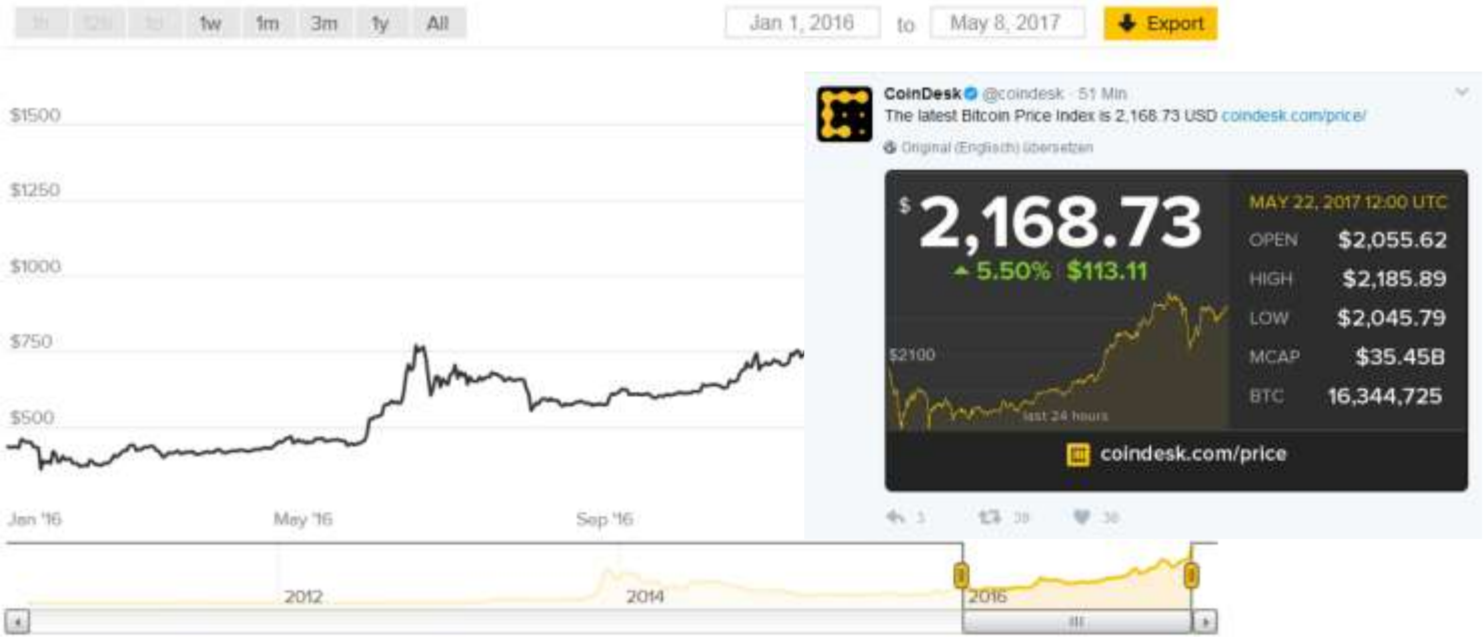
Current Issues

- ➔ Increasing TX backlogs
- ➔ TX confirmation can take hours (instead of 10’)
- ➔ Increasing TX fees
- ➔ «War» between **Bitcoin Core** and **Bitcoin Unlimited** (Scaling Debate)
- ➔ Decreasing market share

Bitcoin Market Share



Bitcoin Price



CoinDesk @coindesk · 51 Min

The latest Bitcoin Price Index is 2,168.73 USD coindesk.com/price/

Original (Englisch) übersetzen

$ 2,168.73

5.50% $113.11

MAY 22, 2017 12:00 UTC

OPEN	$2,055.62
HIGH	$2,185.89
LOW	$2,045.79
MCAP	$35.45B
BTC	16,344,725

coindesk.com/price

Bitcoin Recap

Bitcoin Success

- ➔ Completely decentral currency (no need for central banks)
- ➔ Open Source (GitHub) and Open Data (complete TX history)
- ➔ First successful implementation of any crypto currency
- ➔ «Gold Standard» since 2009
- ➔ Record price levels

Bitcoin Challenges

- ➔ Declining market share
- ➔ Scaling debate/war

Bitcoin Resources

Introduction

What Is Bitcoin?

Bitcoin is a collection of concepts and technologies that form the basis of a digital money ecosystem. Units of currency called bitcoin are used to store and transmit value among participants in the bitcoin network. Bitcoin users communicate with each other using the bitcoin protocol primarily via the Internet, although other transport networks can also be used. The bitcoin

Ethereum

Decentralized Smart Contracts

- 2014 by Vitalik Buterin
- Distributed Turing complete VM
- Open source software 2014
- Is crypto currency too

Ethereum vs Bitcoin

Common Traits

- ➔ Local clients/nodes with complete blockchain (open data)
- ➔ Concepts of addresses, transactions, mining
- ➔ Virtual currency **Ether**
- ➔ Open source

Main Differences

- ➔ **Specification** with different implementations (Bitcoin: single client)
- ➔ **Turing complete scripting** (Bitcoin: very limited scripting)

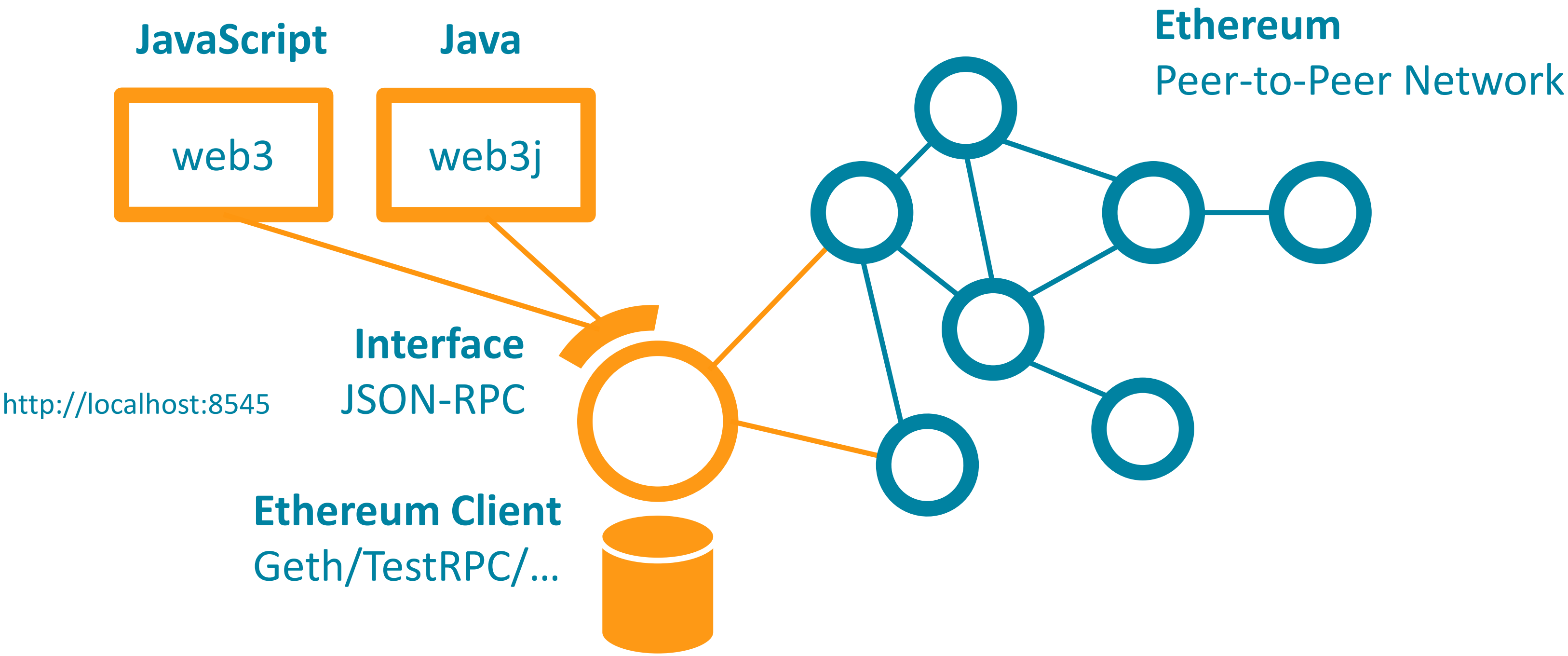
Ethereum as Virtual Currency Platform

- **Ether** is currency unit (1 Ether ~ 87$ 17.05.17)
- **Wei** is smallest denomination (10^{-18} Ether)
- TX mining: Proof of Work (PoW)
- Distributed consensus like Bitcoin (true chain == highest cumulative PoW)

Storing Information

- **Ethereum clients**: Maintain blockchain data + state data
- **State data**: Account balances + nonces
- **Transaction data**: Ether transfers

Ethereum and App Integration



JavaScript

web3

Java

web3j

Interface

JSON-RPC

http://localhost:8545

Ethereum Client

Geth/TestRPC/...

Ethereum

Peer-to-Peer Network

Ethereum Hands-on

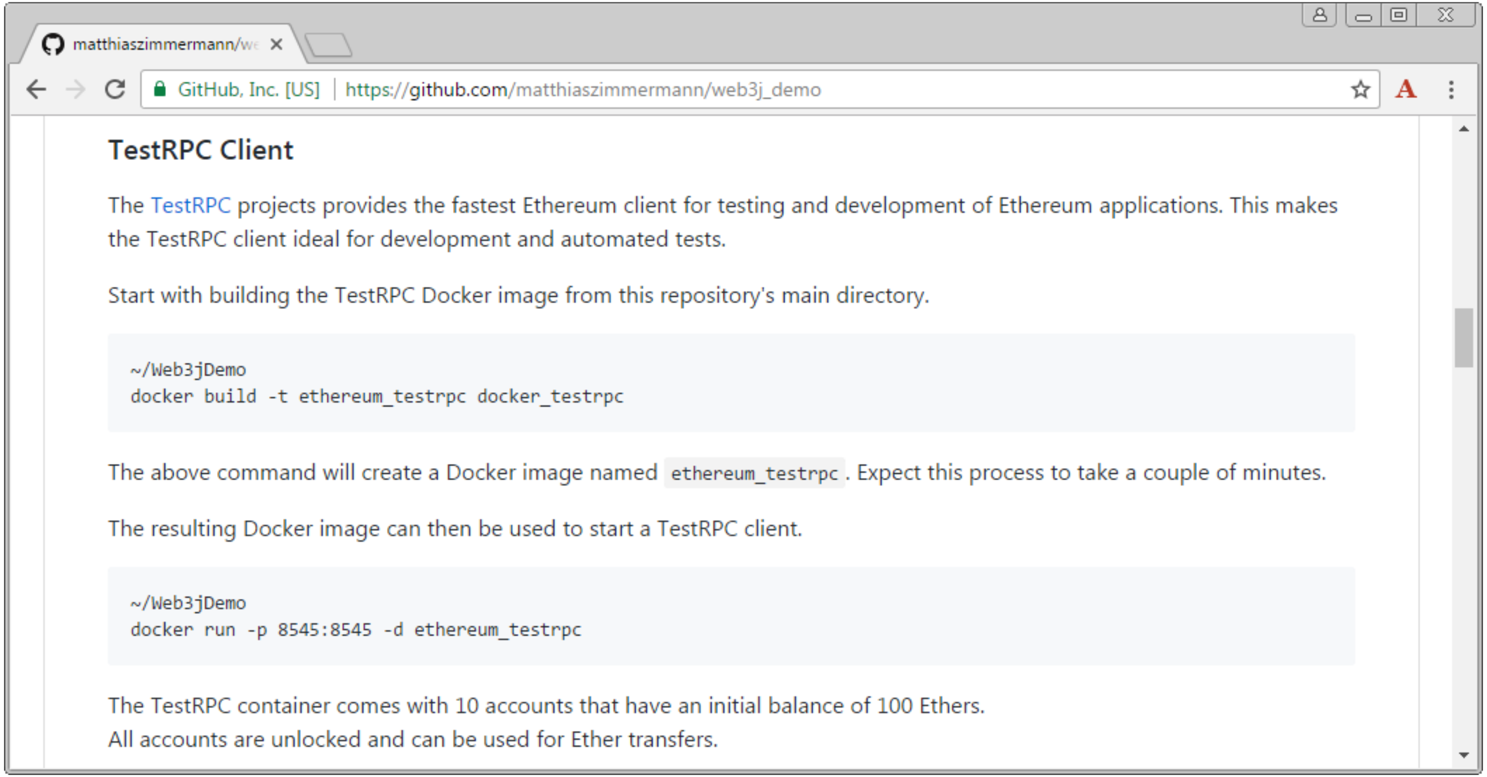
Ideal Development Setup

- ➔ Offline, repeatable, fast
- ➔ Java (this is a JUG talk after all)

We can have all this ☺

- ➔ Docker (repeatable, shareable)
- ➔ TestRPC (offline + local blockchain /w immediate TX confirmations)
- ➔ Web3j (Ethereum Java Library)

TestRPC Docker Image



TestRPC Client

The TestRPC projects provides the fastest Ethereum client for testing and development of Ethereum applications. This makes the TestRPC client ideal for development and automated tests.

Start with building the TestRPC Docker image from this repository's main directory.

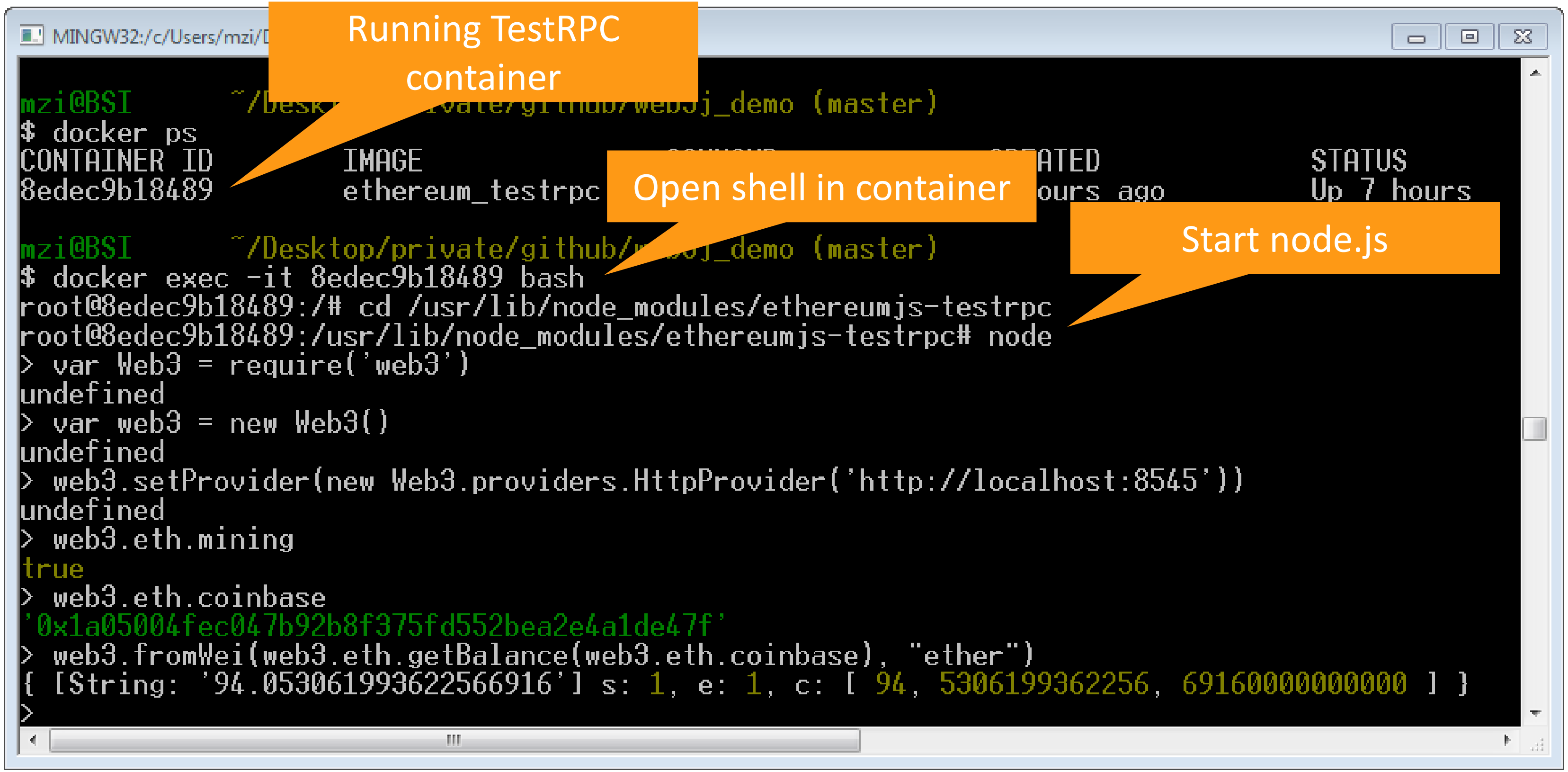
```
~/Web3jDemo
docker build -t ethereum_testrpc docker_testrpc
```

The above command will create a Docker image named `ethereum_testrpc`. Expect this process to take a couple of minutes.

The resulting Docker image can then be used to start a TestRPC client.

```
~/Web3jDemo
docker run -p 8545:8545 -d ethereum_testrpc
```

The TestRPC container comes with 10 accounts that have an initial balance of 100 Ethers.
All accounts are unlocked and can be used for Ether transfers.

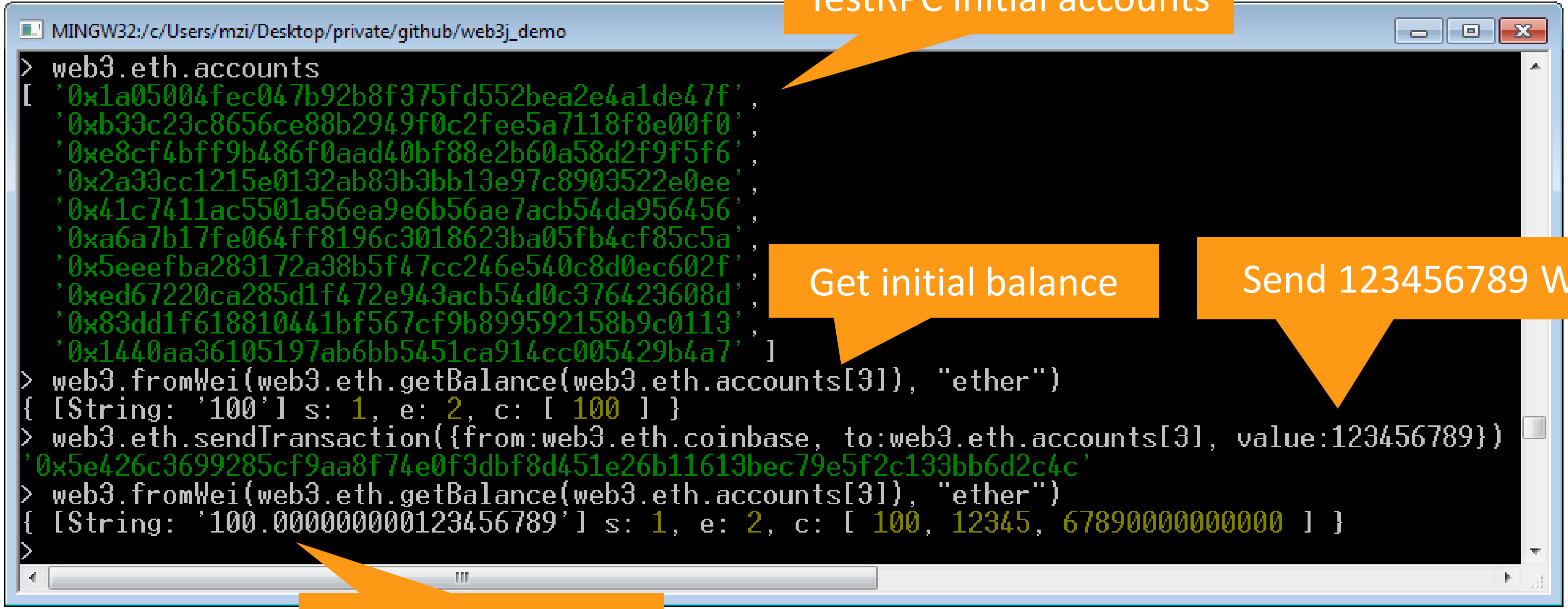


```
mzi@BSI    ~/Desk          ivate/github/web3j_demo (master)
$ docker ps
CONTAINER ID        IMAGE                                      ATED                STATUS
8edec9b18489        ethereum_testrpc                          ours ago            Up 7 hours

mzi@BSI    ~/Desktop/private/github/w      j_demo (master)
$ docker exec -it 8edec9b18489 bash
root@8edec9b18489:/# cd /usr/lib/node_modules/ethereumjs-testrpc
root@8edec9b18489:/usr/lib/node_modules/ethereumjs-testrpc# node
> var Web3 = require('web3')
undefined
> var web3 = new Web3()
undefined
> web3.setProvider(new Web3.providers.HttpProvider('http://localhost:8545'))
undefined
> web3.eth.mining
true
> web3.eth.coinbase
'0x1a05004fec047b92b8f375fd552bea2e4a1de47f'
> web3.fromWei(web3.eth.getBalance(web3.eth.coinbase), "ether")
{ [String: '94.05306199362256916'] s: 1, e: 1, c: [ 94, 5306199362256, 69160000000000 ] }
>
```

Running TestRPC container

Open shell in container

Start node.js



MINGW32:/c/Users/mzi/Desktop/private/github/web3j_demo

```
> web3.eth.accounts
[ '0x1a05004fec047b92b8f375fd552bea2e4a1de47f',
  '0xb33c23c8656ce88b2949f0c2fee5a7118f8e00f0',
  '0xe8cf4bff9b486f0aad40bf88e2b60a58d2f9f5f6',
  '0x2a33cc1215e0132ab83b3bb13e97c8903522e0ee',
  '0x41c7411ac5501a56ea9e6b56ae7acb54da956456',
  '0xa6a7b17fe064ff8196c3018623ba05fb4cf85c5a',
  '0x5eeefba283172a38b5f47cc246e540c8d0ec602f',
  '0xed67220ca285d1f472e943acb54d0c376423608d',
  '0x83dd1f618810441bf567cf9b899592158b9c0113',
  '0x1440aa36105197ab6bb5451ca914cc005429b4a7' ]
> web3.fromWei(web3.eth.getBalance(web3.eth.accounts[3]), "ether")
{ [String: '100'] s: 1, e: 2, c: [ 100 ] }
> web3.eth.sendTransaction({from:web3.eth.coinbase, to:web3.eth.accounts[3], value:123456789})
'0x5e426c3699285cf9aa8f74e0f3dbf8d451e26b11613bec79e5f2c133bb6d2c4c'
> web3.fromWei(web3.eth.getBalance(web3.eth.accounts[3]), "ether")
{ [String: '100.000000000123456789'] s: 1, e: 2, c: [ 100, 12345, 67890000000000 ] }
>
```

TestRPC initial accounts

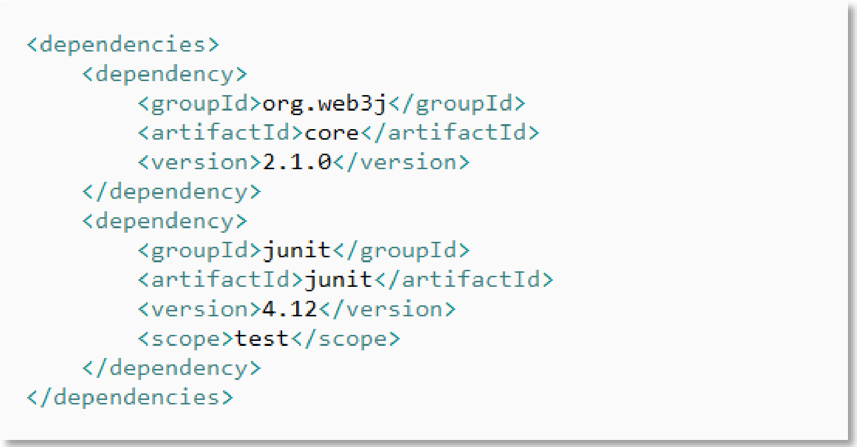
Get initial balance

Send 123456789 Weis

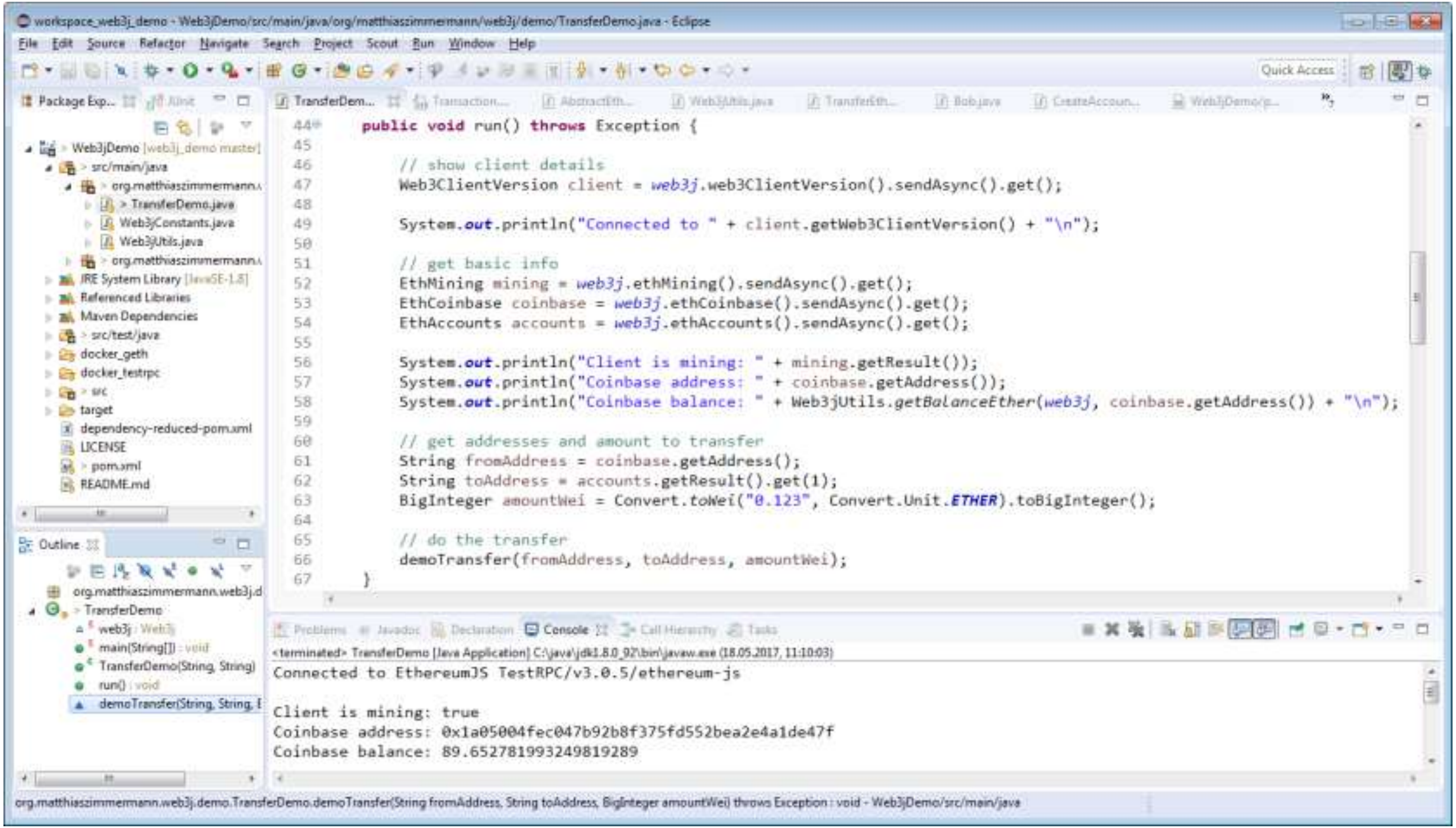
New balance

web3j: Ethereum and Java

- Java implementation of **JSON-RPC client API**
- A couple of other features
- Android support



```
<dependencies>
    <dependency>
        <groupId>org.web3j</groupId>
        <artifactId>core</artifactId>
        <version>2.1.0</version>
    </dependency>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.12</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

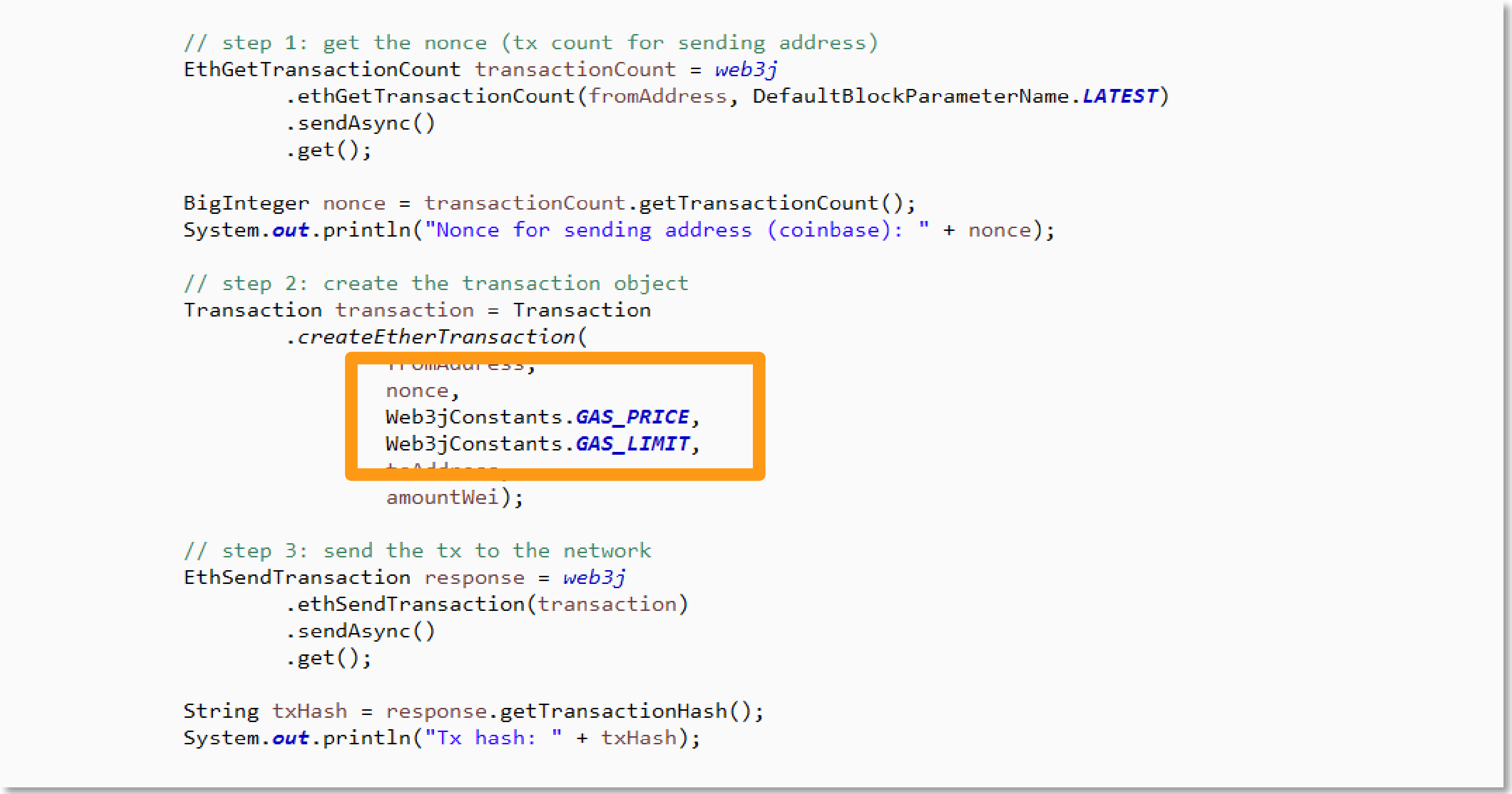

workspace_web3j_demo - Web3jDemo/src/main/java/org/matthiaszimmermann/web3j/demo/TransferDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Scout Run Window Help
Quick Access
Package Exp... JUnit
Web3jDemo [web3j_demo master]
src/main/java
org.matthiaszimmermann.
TransferDemo.java
Web3jConstants.java
Web3jUtils.java
org.matthiaszimmermann.
JRE System Library [JavaSE-1.8]
Referenced Libraries
Maven Dependencies
src/test/java
docker_geth
docker_testrpc
src
target
dependency-reduced-pom.xml
LICENSE
pom.xml
README.md
Outline
org.matthiaszimmermann.web3j.d
TransferDemo
web3j : Web3j
main(String[]) : void
TransferDemo(String, String)
run() : void
demoTransfer(String, String, I
TransferDem... Transaction... AbstractEth... Web3jUtils.java TransferEth... Bob.java CreateAccoun... Web3jDemo/p...
Problems Javadoc Declaration Console Call Hierarchy Tasks

```java
	public void run() throws Exception {

		// show client details
		Web3ClientVersion client = web3j.web3ClientVersion().sendAsync().get();

		System.out.println("Connected to " + client.getWeb3ClientVersion() + "\n");

		// get basic info
		EthMining mining = web3j.ethMining().sendAsync().get();
		EthCoinbase coinbase = web3j.ethCoinbase().sendAsync().get();
		EthAccounts accounts = web3j.ethAccounts().sendAsync().get();

		System.out.println("Client is mining: " + mining.getResult());
		System.out.println("Coinbase address: " + coinbase.getAddress());
		System.out.println("Coinbase balance: " + Web3jUtils.getBalanceEther(web3j, coinbase.getAddress()) + "\n");

		// get addresses and amount to transfer
		String fromAddress = coinbase.getAddress();
		String toAddress = accounts.getResult().get(1);
		BigInteger amountWei = Convert.toWei("0.123", Convert.Unit.ETHER).toBigInteger();

		// do the transfer
		demoTransfer(fromAddress, toAddress, amountWei);
	}
```

```
<terminated> TransferDemo [Java Application] C:\java\jdk1.8.0_92\bin\javaw.exe (18.05.2017, 11:10:03)
Connected to EthereumJS TestRPC/v3.0.5/ethereum-js

Client is mining: true
Coinbase address: 0x1a05004fec047b92b8f375fd552bea2e4a1de47f
Coinbase balance: 89.652781993249819289
```

org.matthiaszimmermann.web3j.demo.TransferDemo.demoTransfer(String fromAddress, String toAddress, BigInteger amountWei) throws Exception : void - Web3jDemo/src/main/java

web3j: Creating and Sending TX



```
// step 1: get the nonce (tx count for sending address)
EthGetTransactionCount transactionCount = web3j
        .ethGetTransactionCount(fromAddress, DefaultBlockParameterName.LATEST)
        .sendAsync()
        .get();

BigInteger nonce = transactionCount.getTransactionCount();
System.out.println("Nonce for sending address (coinbase): " + nonce);

// step 2: create the transaction object
Transaction transaction = Transaction
        .createEtherTransaction(
                fromAddress,
                nonce,
                Web3jConstants.GAS_PRICE,
                Web3jConstants.GAS_LIMIT,
                toAddress,
                amountWei);

// step 3: send the tx to the network
EthSendTransaction response = web3j
        .ethSendTransaction(transaction)
        .sendAsync()
        .get();

String txHash = response.getTransactionHash();
System.out.println("Tx hash: " + txHash);
```

Ethereum Accounts and Nonces

How nonces are used

- Each account has a **nonce** value (account state data)
- Accounts start with nonce value 0
- TX: includes **sender address** and its **nonce** value
- TX can only be mined if:
  - Account has sufficient funds
  - TX nonce == current account nonce
- If TX is mined successfully: Nonce increased by 1

Ethereum, Gas and TX Fees

What is gas?

- ➔ Special unit to pay fees to mining nodes
- ➔ Gas has price in Ethers (decouples computing costs and Ether price)

What fees?

- ➔ Computations performed by **Ethereum Virtual Machine** (EVM)
- ➔ EVM is working as long as there is gas
- ➔ **Example 1**: **SHA3** computation costs 30 gas
- ➔ **Example 2**: EVM always terminates (stays in **infinite loop** until gas runs out)

Smart Contracts

Ethereum Smart Contracts

What is a Smart Contract?

- ➔ Piece of (byte) code
- ➔ Is executed by the Ethereum Virtual Machine (EVM)
- ➔ Has an owner
- ➔ Has a life cycle
- ➔ Might have some purpose

Examples

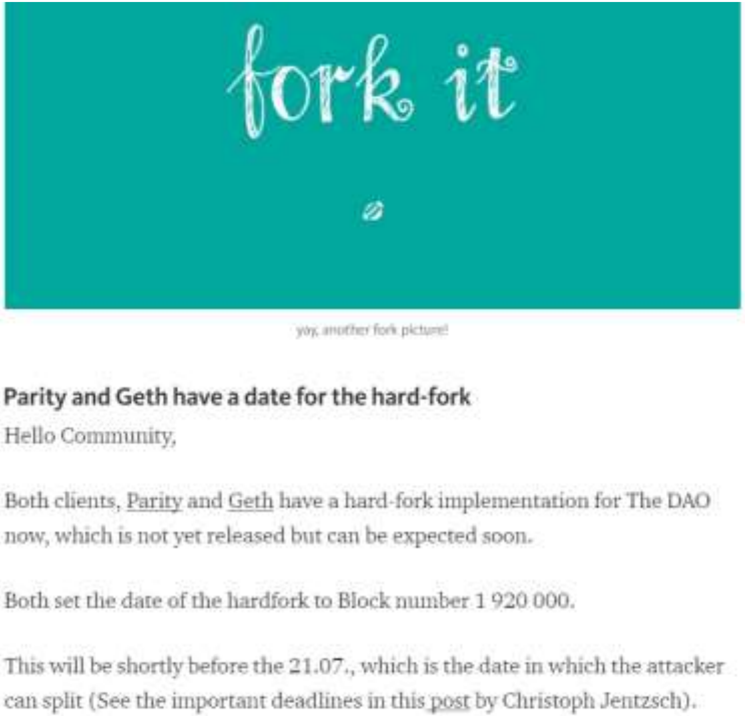
1. The DAO
2. Flight delay insurance
3. «Truly» autonomous cars

The DAO (2016)

- ➔ Distributed venture capital fund
- ➔ Amount raised $150,000,000
- ➔ Largest crowdfunding project
- ➔ Successfully «attacked»

Attack Result: Ethereum hard fork

- ▪ ETH «true» blockchain
- ▪ ETC «forked» blockchain



yay, another fork picture!

Parity and Geth have a date for the hard-fork

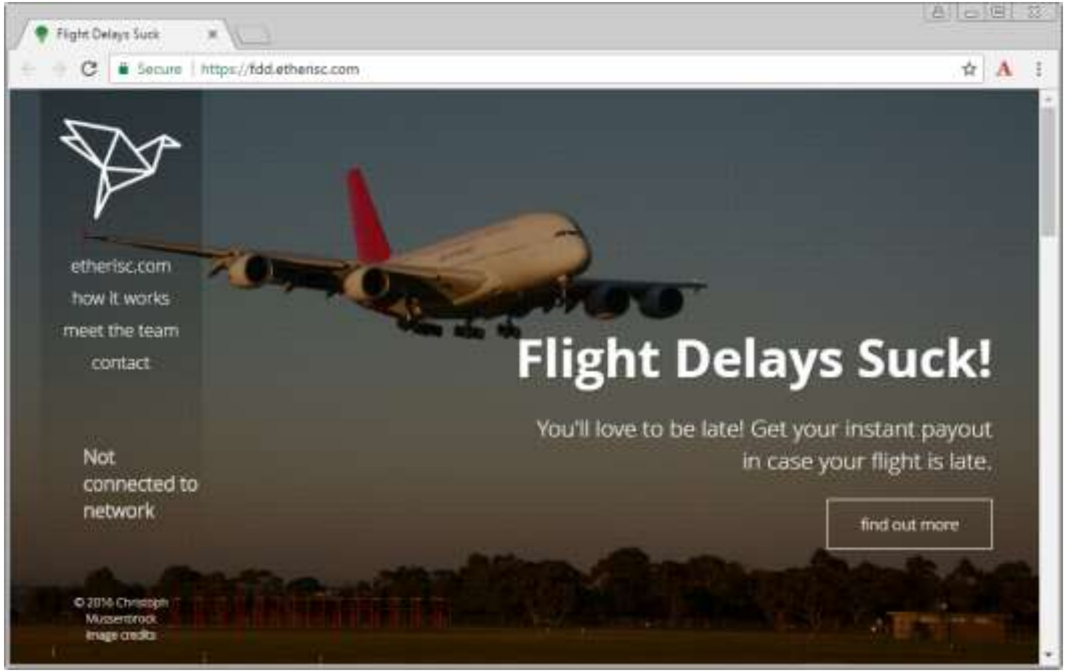
Hello Community,

Both clients, Parity and Geth have a hard-fork implementation for The DAO now, which is not yet released but can be expected soon.

Both set the date of the hardfork to Block number 1 920 000.

This will be shortly before the 21.07., which is the date in which the attacker can split (See the important deadlines in this post by Christoph Jentzsch).

Flight Delay App

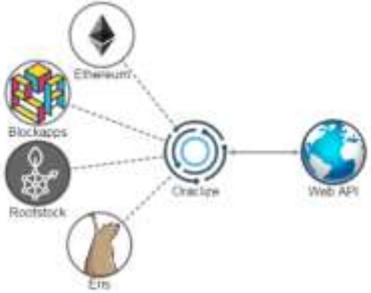


Flight Delays Suck!

You'll love to be late! Get your instant payout in case your flight is late.

Not connected to network

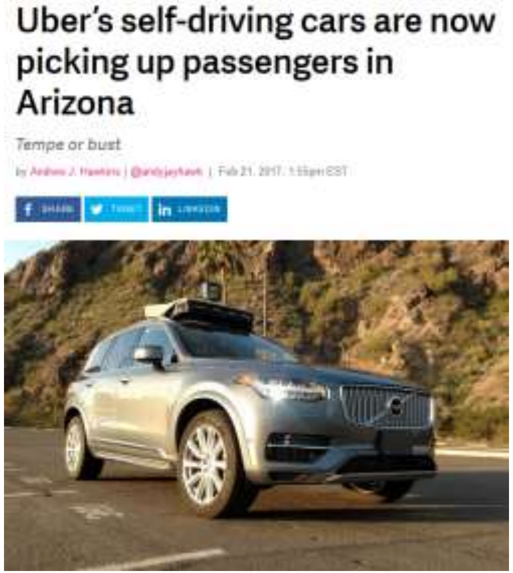
© 2016 Christoph Mussenbrock
image credits



FlightStats is the leader in flight data services and applications to customers serving the global travel industry

→ Involves Oraclize service to access flightstats.com

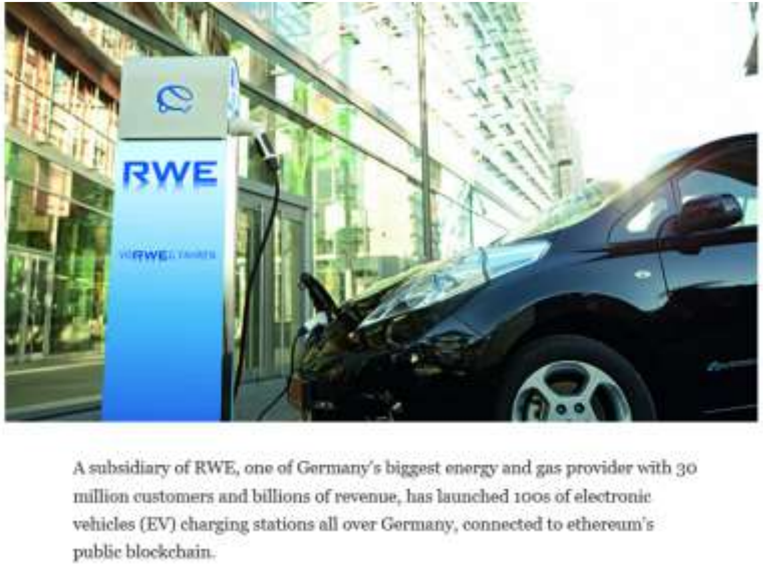
«Truly» Autonomous Cars



Uber's self-driving cars are now picking up passengers in Arizona

Tempe or bust

by Andrew J. Hawkins | @andyjayhawk | Feb 21, 2017, 1:55pm EST



A subsidiary of RWE, one of Germany's biggest energy and gas provider with 30 million customers and billions of revenue, has launched 100s of electronic vehicles (EV) charging stations all over Germany, connected to ethereum's public blockchain.

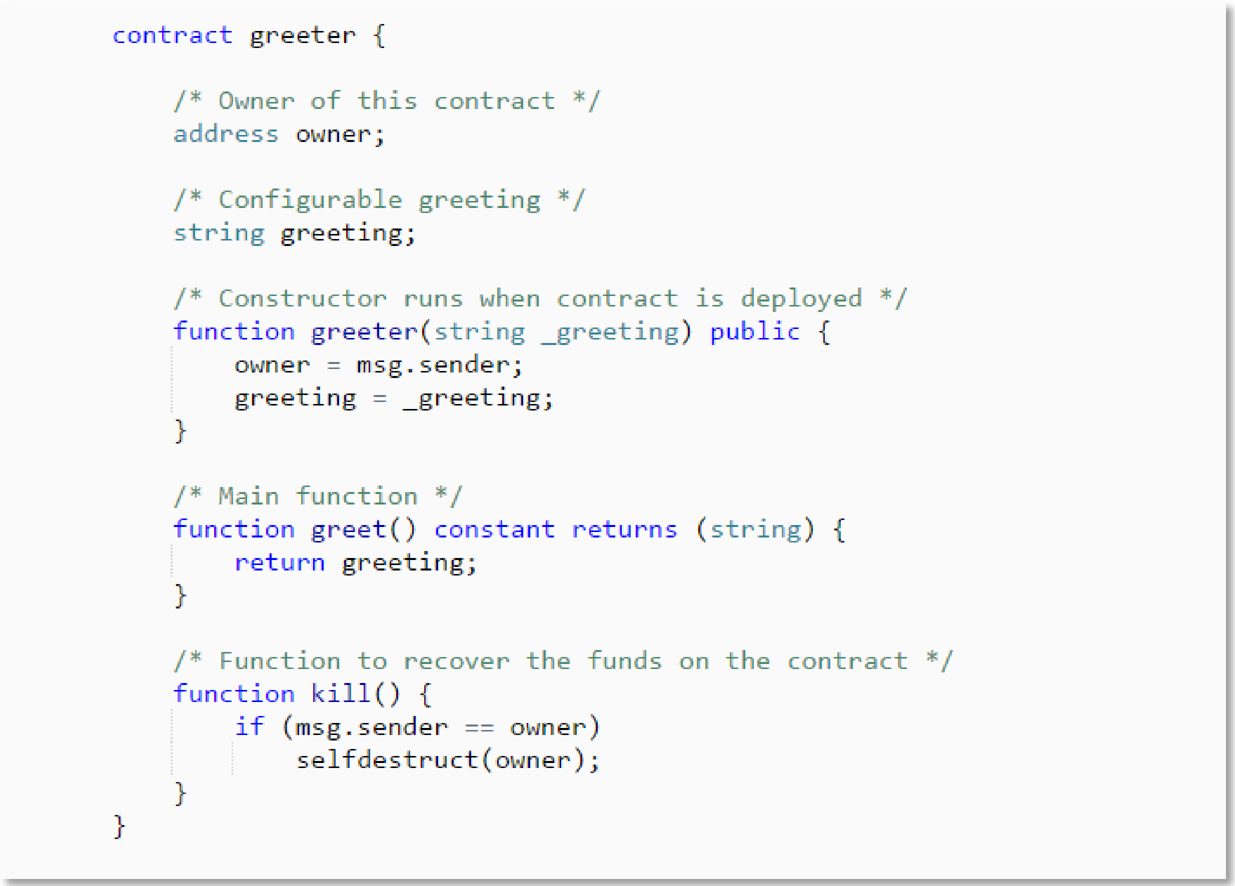
- ➔ Smart contract to ordering vehicle to transport goods/people
- ➔ Smart contract to pay for energy/services

Smart Contracts Life Cycle

Deploying and using Smart Contracts

1. Write contract in high level language (eg. **Solidity**)
2. Compile contract to **EVM byte-code**
3. Pack byte code into a **contract creation TX** and sent to the network
4. The TX gets ist own contract account
5. Contract account has address, balance, nonce and holds byte code
6. Invoke methods using calls (free) or transactions (cost gas)

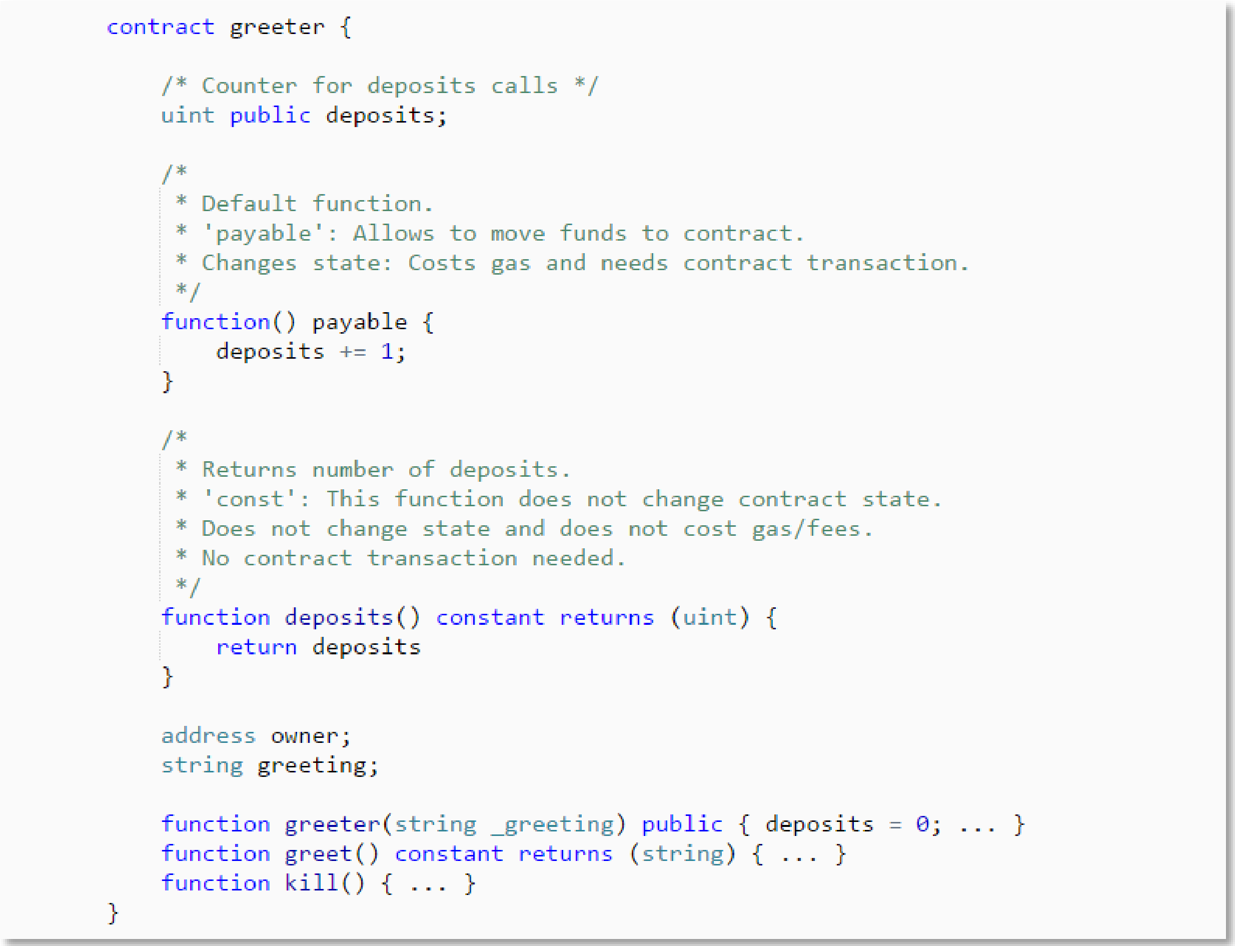
«Hello World»   (greeter.sol)



```
contract greeter {

    /* Owner of this contract */
    address owner;

    /* Configurable greeting */
    string greeting;

    /* Constructor runs when contract is deployed */
    function greeter(string _greeting) public {
        owner = msg.sender;
        greeting = _greeting;
    }

    /* Main function */
    function greet() constant returns (string) {
        return greeting;
    }

    /* Function to recover the funds on the contract */
    function kill() {
        if (msg.sender == owner)
            selfdestruct(owner);
    }
}
```

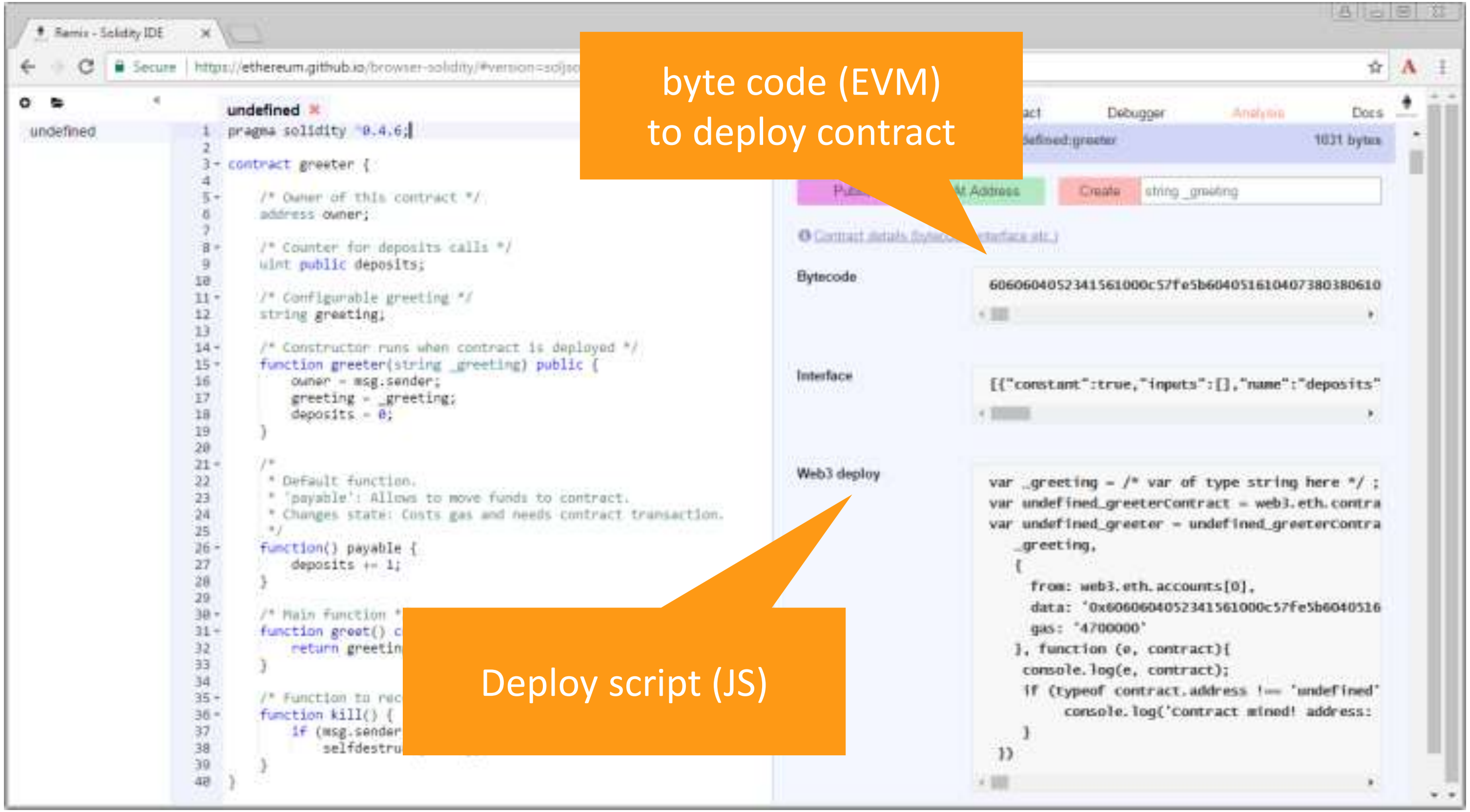
greeter.sol ++

- additional state
- ‘payable’



```
contract greeter {

    /* Counter for deposits calls */
    uint public deposits;

    /*
     * Default function.
     * 'payable': Allows to move funds to contract.
     * Changes state: Costs gas and needs contract transaction.
     */
    function() payable {
        deposits += 1;
    }

    /*
     * Returns number of deposits.
     * 'const': This function does not change contract state.
     * Does not change state and does not cost gas/fees.
     * No contract transaction needed.
     */
    function deposits() constant returns (uint) {
        return deposits
    }

    address owner;
    string greeting;

    function greeter(string _greeting) public { deposits = 0; ... }
    function greet() constant returns (string) { ... }
    function kill() { ... }
}
```

Solidity Compiler (online)



```
pragma solidity ^0.4.6;

contract greeter {

    /* Owner of this contract */
    address owner;

    /* Counter for deposits calls */
    uint public deposits;

    /* Configurable greeting */
    string greeting;

    /* Constructor runs when contract is deployed */
    function greeter(string _greeting) public {
        owner = msg.sender;
        greeting = _greeting;
        deposits = 0;
    }

    /*
     * Default function.
     * 'payable': Allows to move funds to contract.
     * Changes state: Costs gas and needs contract transaction.
     */
    function() payable {
        deposits += 1;
    }

    /* Main function *
    function greet() c
        return greetin
    }

    /* Function to rec
    function kill() {
        if (msg.sender
            selfdestru
    }
}
```

Bytecode

```
60606040523415610000c57fe5b604051610407380380610
```

Interface

```
[{"constant":true,"inputs":[],"name":"deposits"
```

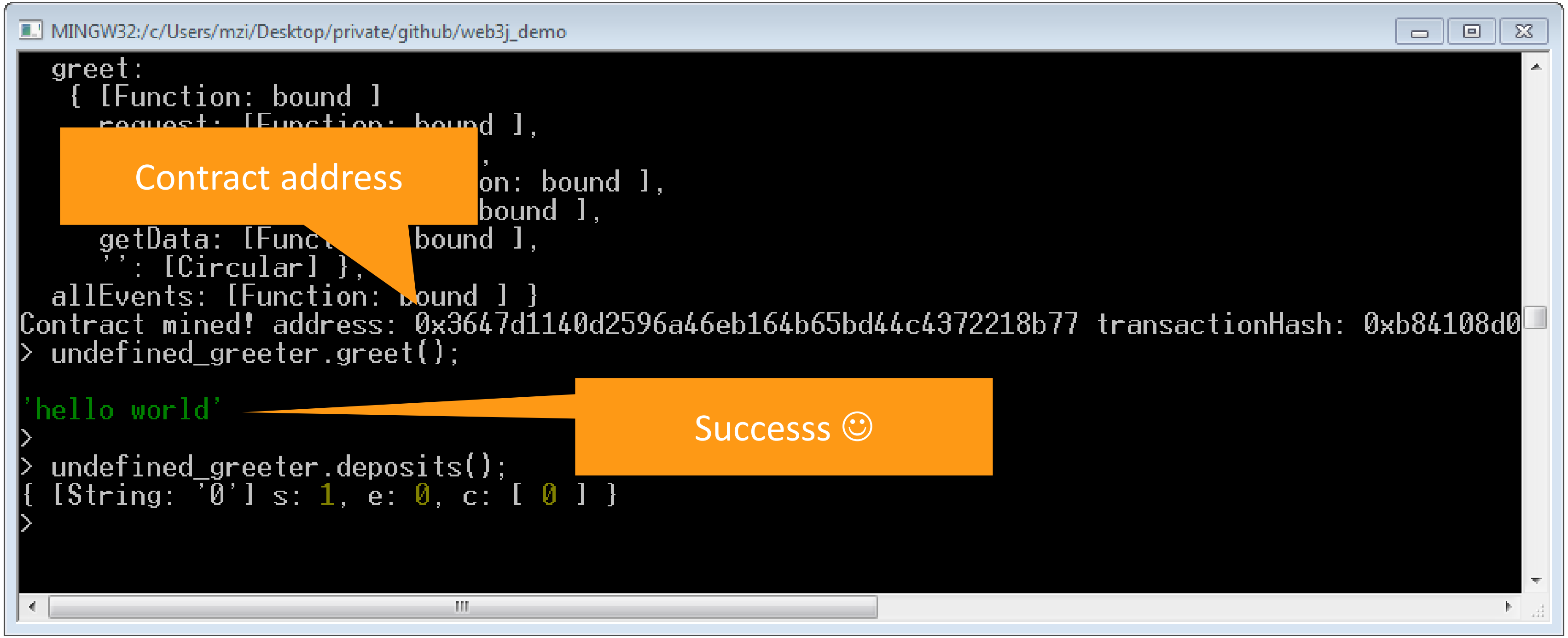
Web3 deploy

```
var _greeting = /* var of type string here */ ;
var undefined_greeterContract = web3.eth.contra
var undefined_greeter = undefined_greeterContra
   _greeting,
   {
     from: web3.eth.accounts[0],
     data: '0x60606040523415610000c57fe5b6040516
     gas: '4700000'
   }, function (e, contract){
    console.log(e, contract);
    if (typeof contract.address !== 'undefined'
         console.log('contract mined! address:
    }
 })
```

byte code (EVM)
to deploy contract

Deploy script (JS)

Deploy (Console)



MINGW32:/c/Users/mzi/Desktop/private/github/web3j_demo

```
  greet:
   { [Function: bound ]
     request: [Function: bound ],
                                ,
                               on: bound ],
                               bound ],
     getData: [Func            bound ],
     '': [Circular] },
  allEvents: [Function:  ound ] }
Contract mined! address: 0x3647d1140d2596a46eb164b65bd44c4372218b77 transactionHash: 0xb84108d0
> undefined_greeter.greet();

'hello world'
>
> undefined_greeter.deposits();
{ [String: '0'] s: 1, e: 0, c: [ 0 ] }
>
```

Contract address

Successs ☺

web3j: greeter.sol → Greeter.java

From Solidity to Java Contract Class

1. Compile `greeter.sol` (e.g using online compiler)

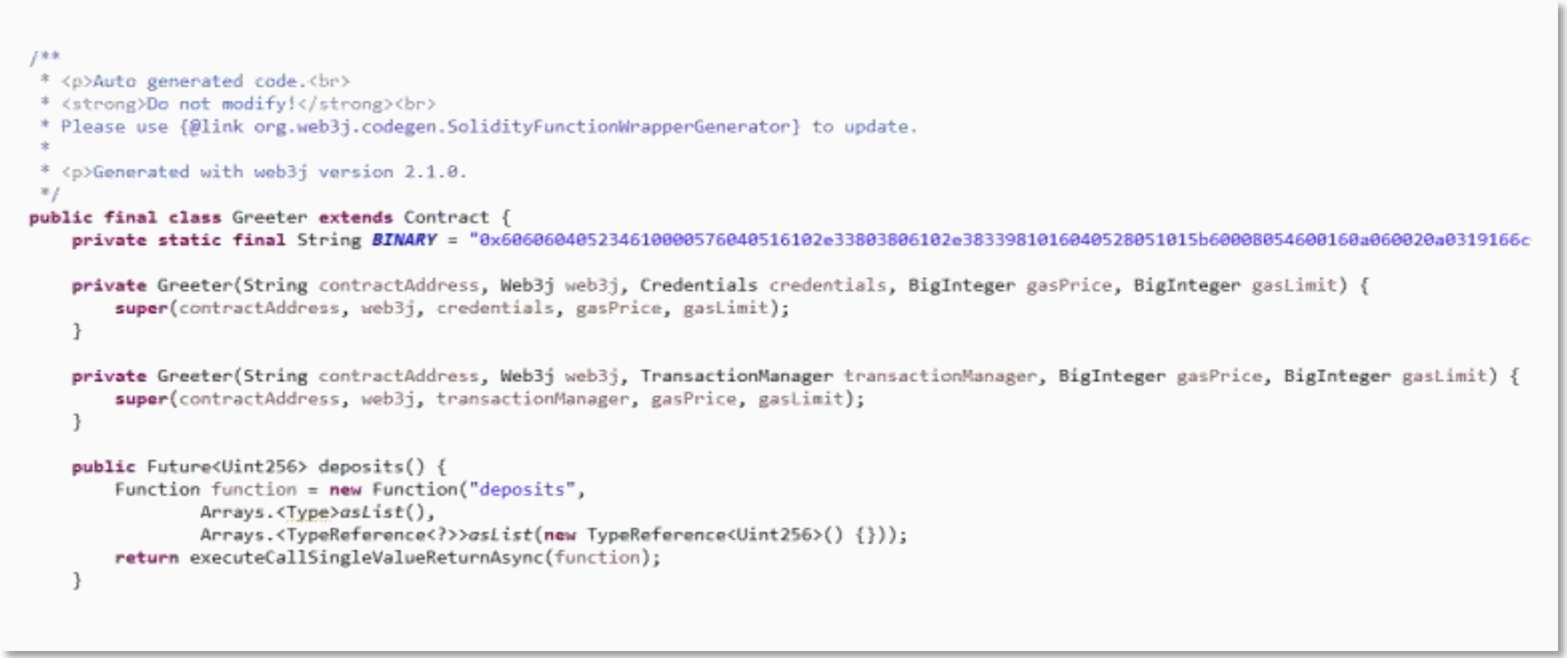
   ➔ `greeter.bin, greeter.abi`

2. Create contract wrapper class (use Web3j command line tool)

   ➔ `Greeter.java`

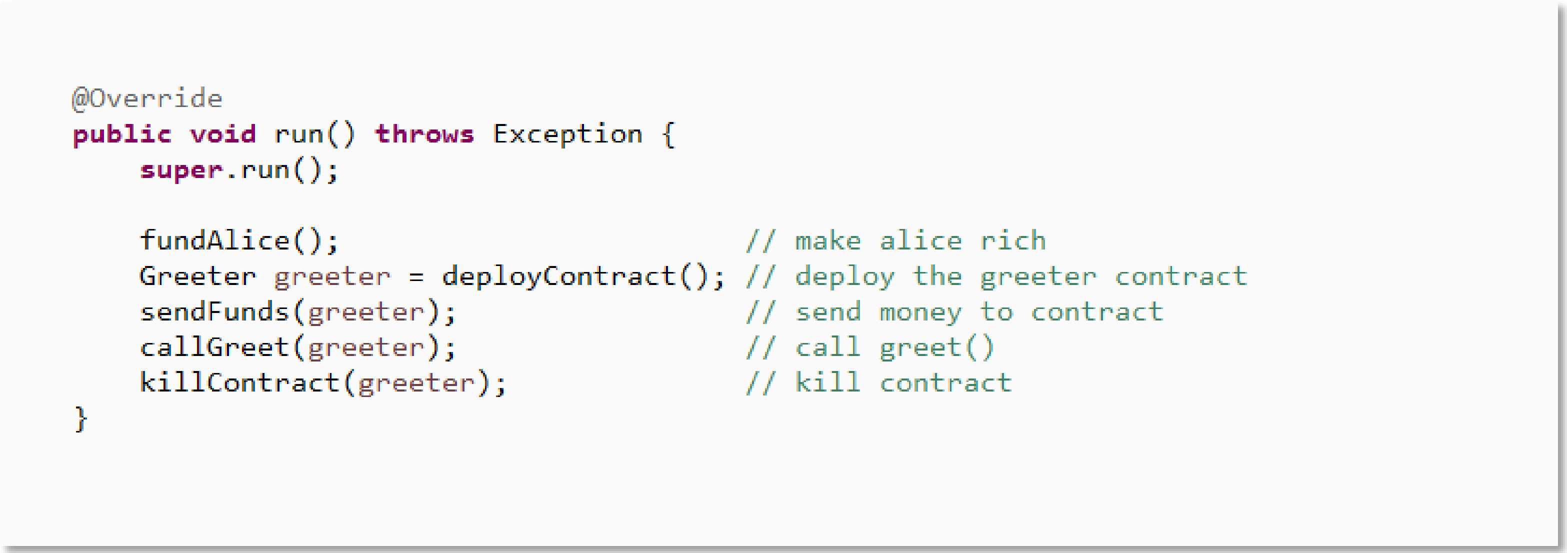
1. Use `Greeter.java` in your Java code

Generated Contract Wrapper



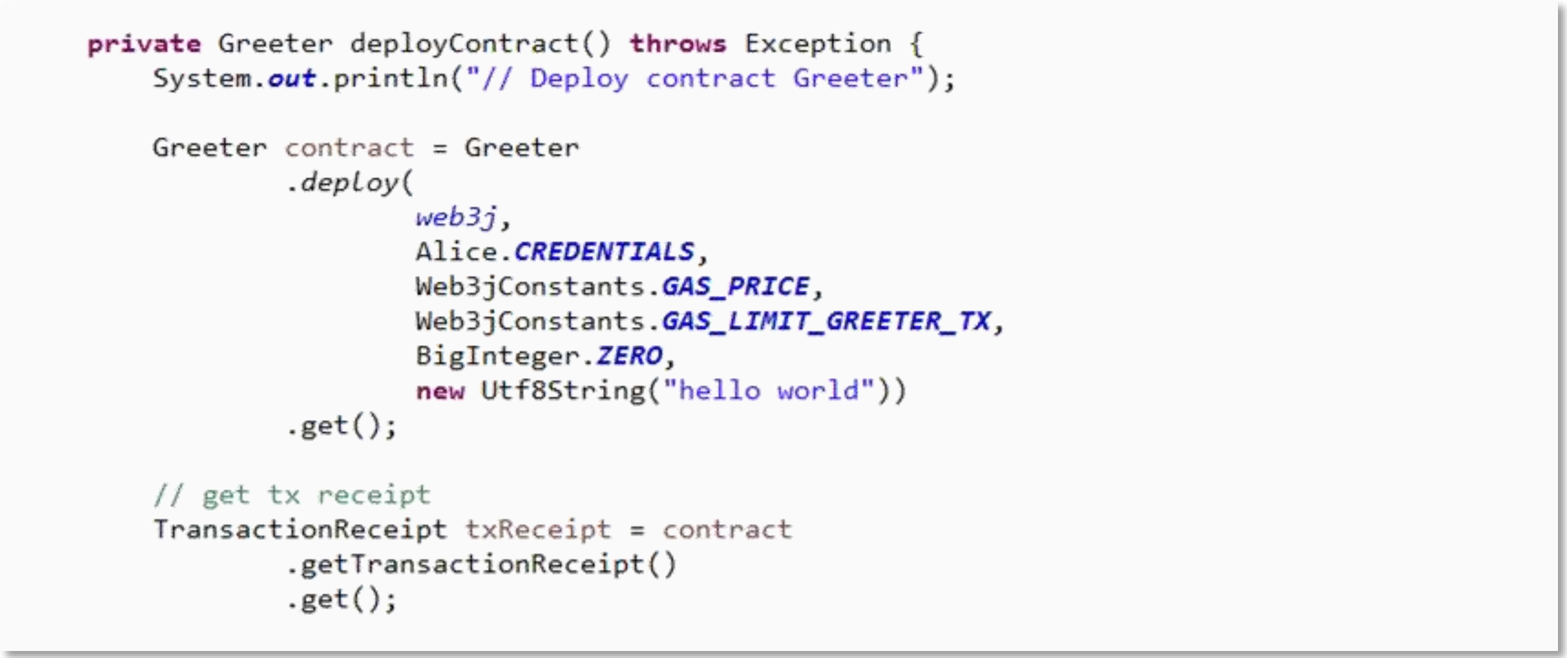
```
/**
 * <p>Auto generated code.<br>
 * <strong>Do not modify!</strong><br>
 * Please use {@link org.web3j.codegen.SolidityFunctionWrapperGenerator} to update.
 *
 * <p>Generated with web3j version 2.1.0.
 */
public final class Greeter extends Contract {
    private static final String BINARY = "0x6060604052346100005760405161 02e33803806102e38339810160405280510151015b600080546001 60a060020a0319166c

    private Greeter(String contractAddress, Web3j web3j, Credentials credentials, BigInteger gasPrice, BigInteger gasLimit) {
        super(contractAddress, web3j, credentials, gasPrice, gasLimit);
    }

    private Greeter(String contractAddress, Web3j web3j, TransactionManager transactionManager, BigInteger gasPrice, BigInteger gasLimit) {
        super(contractAddress, web3j, transactionManager, gasPrice, gasLimit);
    }

    public Future<Uint256> deposits() {
        Function function = new Function("deposits",
                Arrays.<Type>asList(),
                Arrays.<TypeReference<?>>asList(new TypeReference<Uint256>() {}));
        return executeCallSingleValueReturnAsync(function);
    }
```

Live Cycle in Java

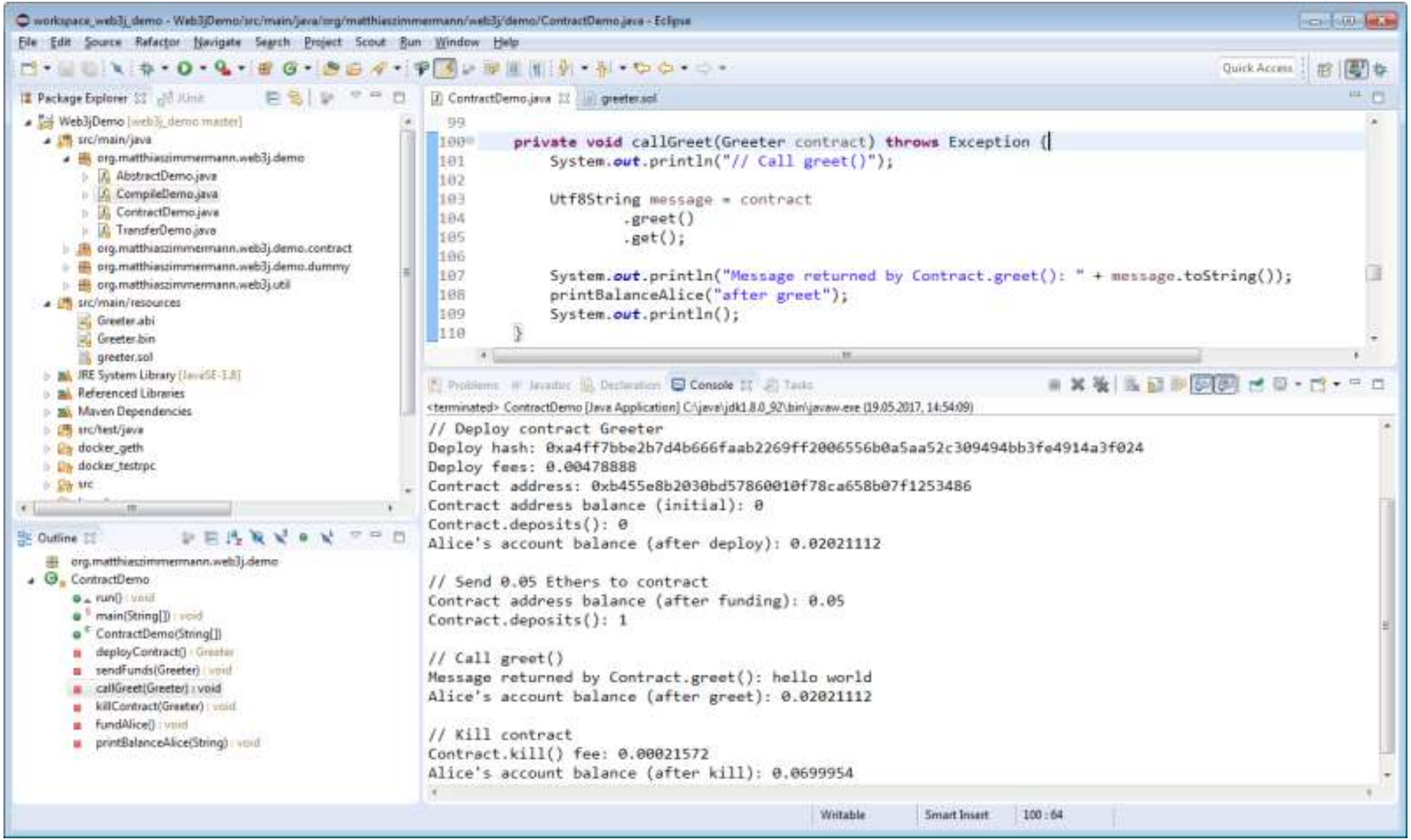


```java
@Override
public void run() throws Exception {
    super.run();

    fundAlice();                        // make alice rich
    Greeter greeter = deployContract(); // deploy the greeter contract
    sendFunds(greeter);                 // send money to contract
    callGreet(greeter);                 // call greet()
    killContract(greeter);              // kill contract
}
```

Deployment



```
private Greeter deployContract() throws Exception {
    System.out.println("// Deploy contract Greeter");

    Greeter contract = Greeter
            .deploy(
                    web3j,
                    Alice.CREDENTIALS,
                    Web3jConstants.GAS_PRICE,
                    Web3jConstants.GAS_LIMIT_GREETER_TX,
                    BigInteger.ZERO,
                    new Utf8String("hello world"))
            .get();

    // get tx receipt
    TransactionReceipt txReceipt = contract
            .getTransactionReceipt()
            .get();
```


workspace_web3j_demo - Web3jDemo/src/main/java/org/matthiaszimmermann/web3j/demo/ContractDemo.java - Eclipse
File Edit Source Refactor Navigate Search Project Scout Run Window Help
Quick Access
Package Explorer
JUnit
Web3jDemo [web3j_demo master]
src/main/java
org.matthiaszimmermann.web3j.demo
AbstractDemo.java
CompileDemo.java
ContractDemo.java
TransferDemo.java
org.matthiaszimmermann.web3j.demo.contract
org.matthiaszimmermann.web3j.demo.dummy
org.matthiaszimmermann.web3j.util
src/main/resources
Greeter.abi
Greeter.bin
greeter.sol
JRE System Library [JavaSE-1.8]
Referenced Libraries
Maven Dependencies
src/test/java
docker_geth
docker_testrpc
src
Outline
org.matthiaszimmermann.web3j.demo
ContractDemo
run() : void
main(String[]) : void
ContractDemo(String[])
deployContract() : Greeter
sendFunds(Greeter) : void
callGreet(Greeter) : void
killContract(Greeter) : void
fundAlice() : void
printBalanceAlice(String) : void
ContractDemo.java
greeter.sol
Problems
Javadoc
Declaration
Console
Tasks
Writable
Smart Insert
100 : 64

```java
    private void callGreet(Greeter contract) throws Exception {
        System.out.println("// Call greet()");

        Utf8String message = contract
                .greet()
                .get();

        System.out.println("Message returned by Contract.greet(): " + message.toString());
        printBalanceAlice("after greet");
        System.out.println();
    }
```

<terminated> ContractDemo [Java Application] C:\java\jdk1.8.0_92\bin\javaw.exe (19.05.2017, 14:54:09)

```
// Deploy contract Greeter
Deploy hash: 0xa4ff7bbe2b7d4b666faab2269ff2006556b0a5aa52c309494bb3fe4914a3f024
Deploy fees: 0.00478888
Contract address: 0xb455e8b2030bd57860010f78ca658b07f1253486
Contract address balance (initial): 0
Contract.deposits(): 0
Alice's account balance (after deploy): 0.02021112

// Send 0.05 Ethers to contract
Contract address balance (after funding): 0.05
Contract.deposits(): 1

// Call greet()
Message returned by Contract.greet(): hello world
Alice's account balance (after greet): 0.02021112

// Kill contract
Contract.kill() fee: 0.00021572
Alice's account balance (after kill): 0.0699954
```

Trading-Network Demo

Ethereum, web3j, Eclipse Scout

Trading Network Demo

Use Case

➔ **Currency Hedging**: Buy orders and Sell orders (€ / US$)

➔ **Classical Business App**

- **Idendity Management** for mapping real persons ←→BC addresses
- **User Interface**

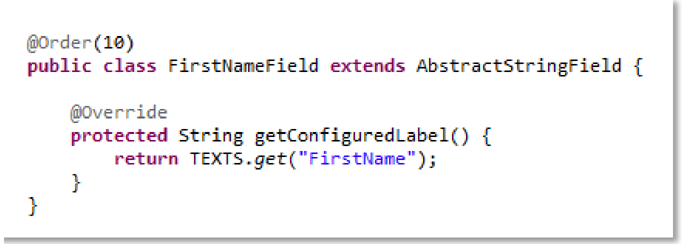
➔ **Blockchain Benefits**

- **Efficiency**: No central organization/infrastructure
- **Trust**: Tampering-proof ledger, trust by blockchain

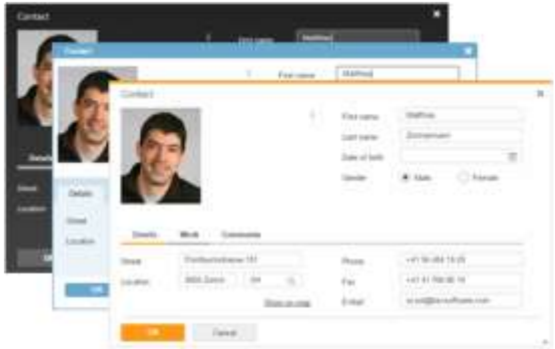
Eclipse Scout

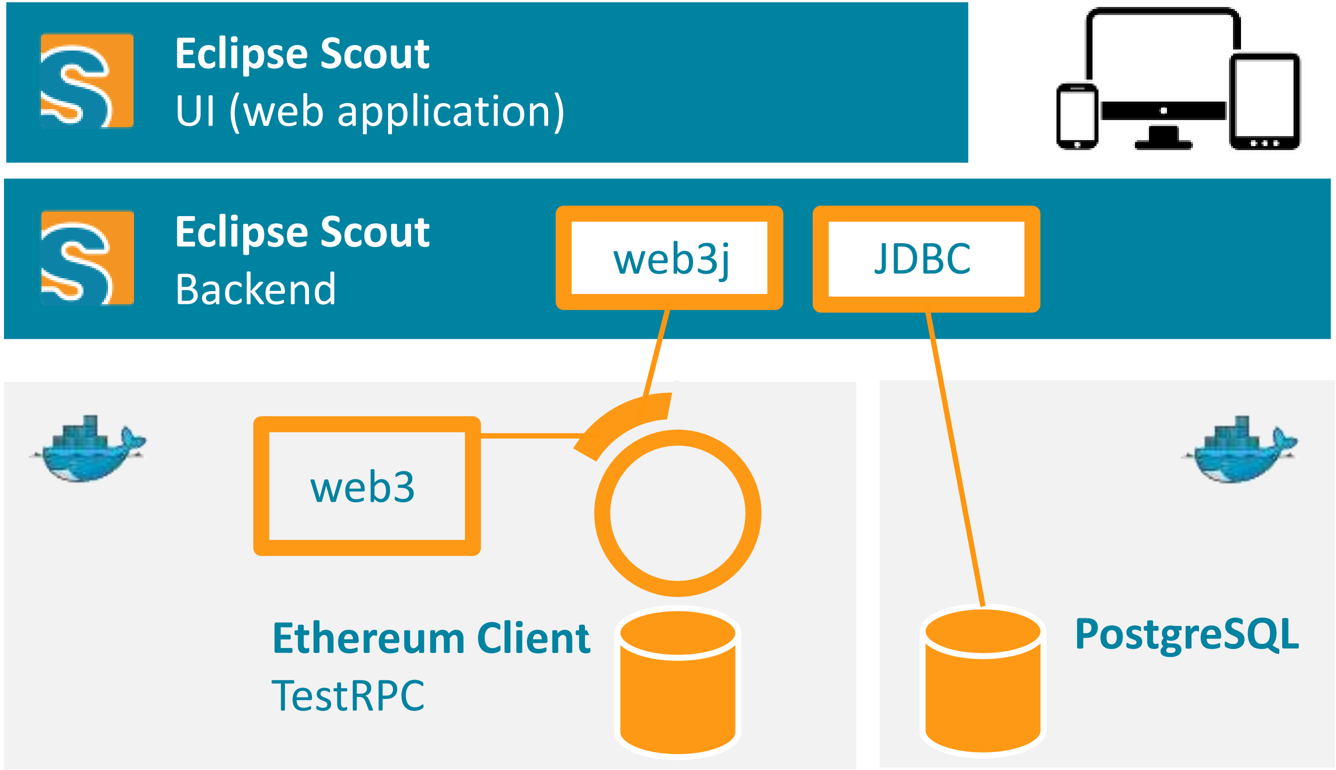
Business Application Framework

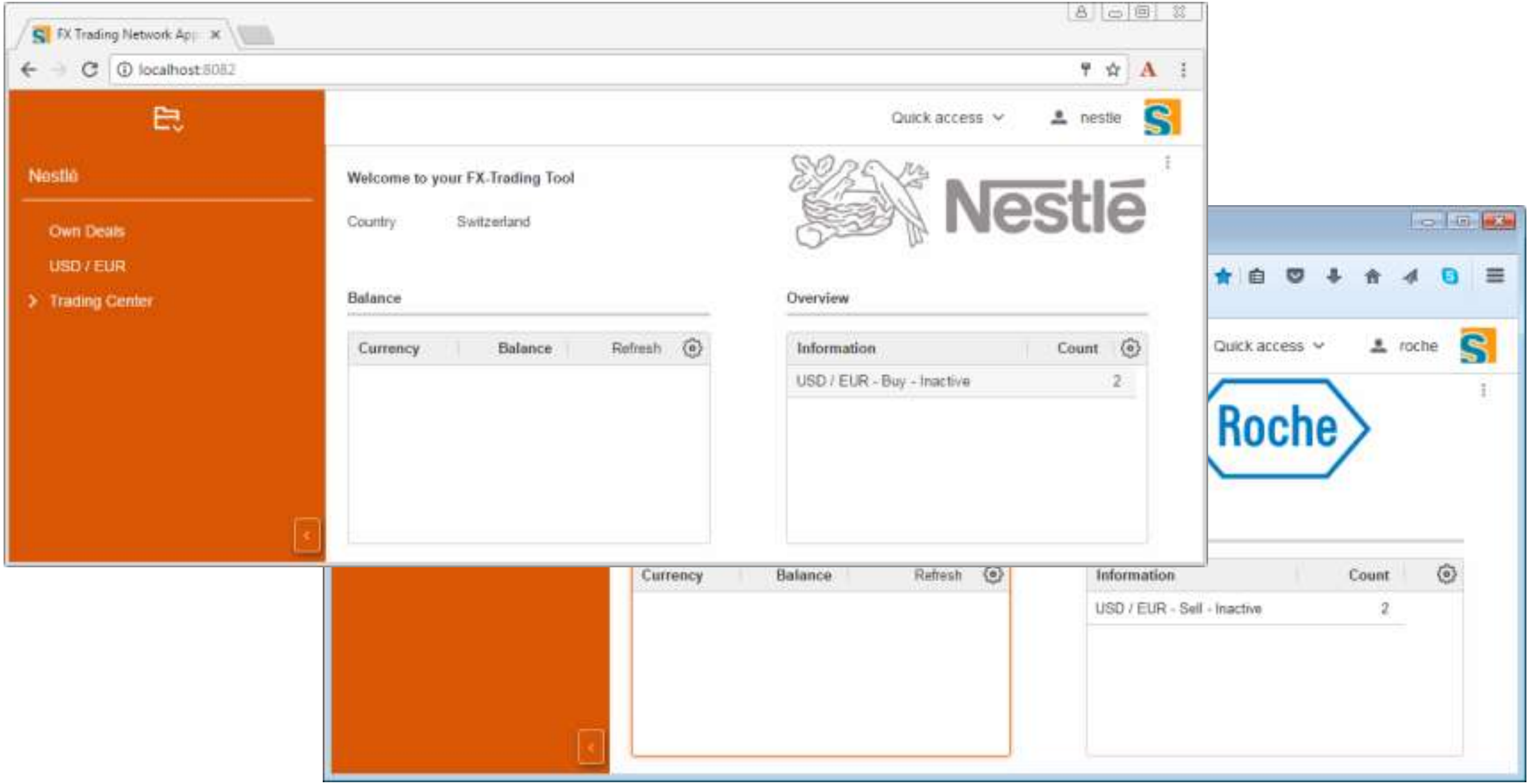
- ➔ Plain Java Application Model
- ➔ Multi-Device Support
- ➔ HTML5/CSS3 Rendering



```java
@Order(10)
public class FirstNameField extends AbstractStringField {

    @Override
    protected String getConfiguredLabel() {
        return TEXTS.get("FirstName");
    }
}
```




Eclipse Scout
UI (web application)
Eclipse Scout
Backend
web3j
JDBC
web3
Ethereum Client
TestRPC
PostgreSQL



Welcome to your FX-Trading Tool

Country Switzerland

Balance

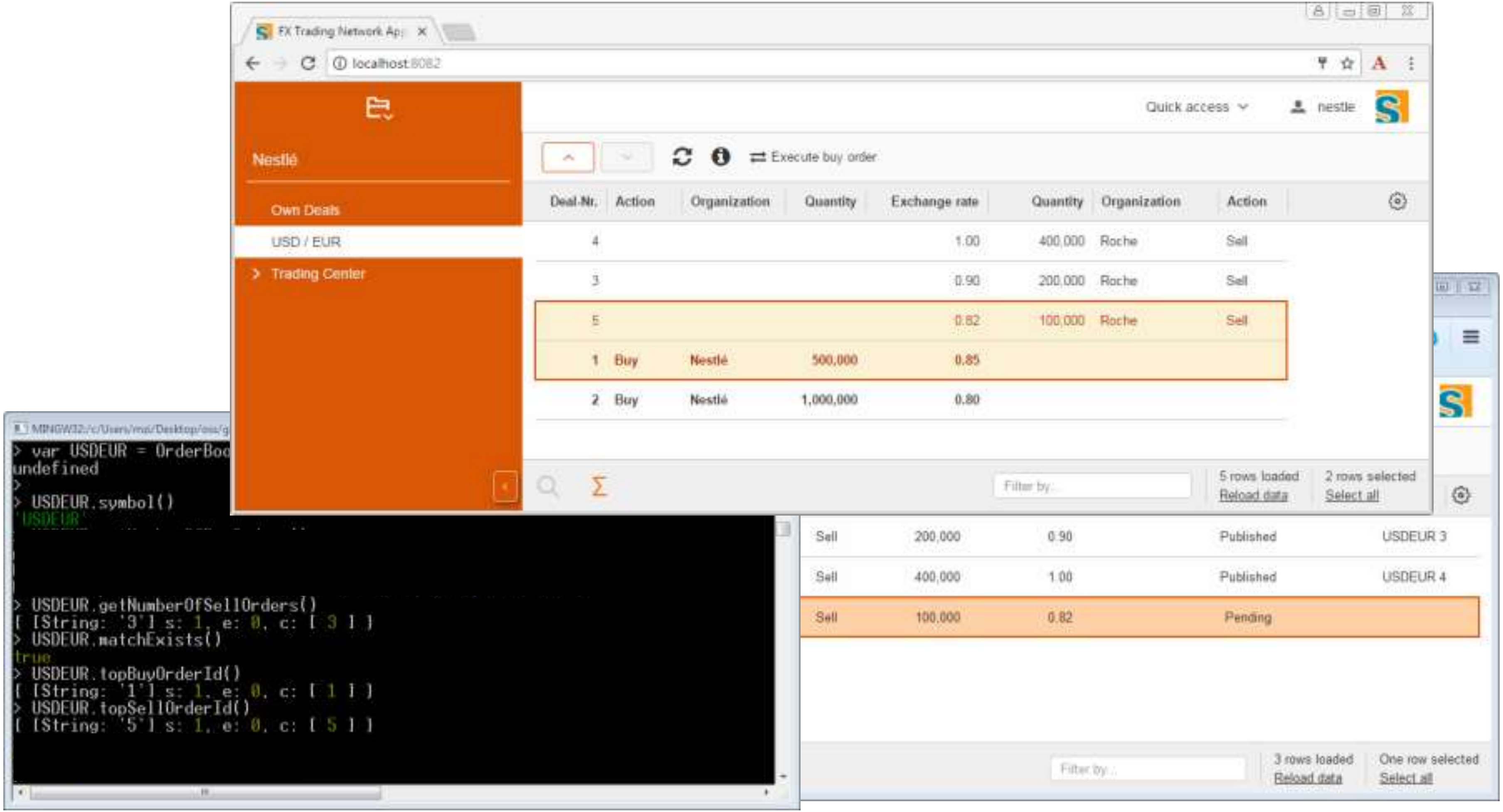
Currency	Balance	Refresh

Overview

Information	Count
USD / EUR - Buy - Inactive	2

Currency	Balance	Refresh

Information	Count
USD / EUR - Sell - Inactive	2

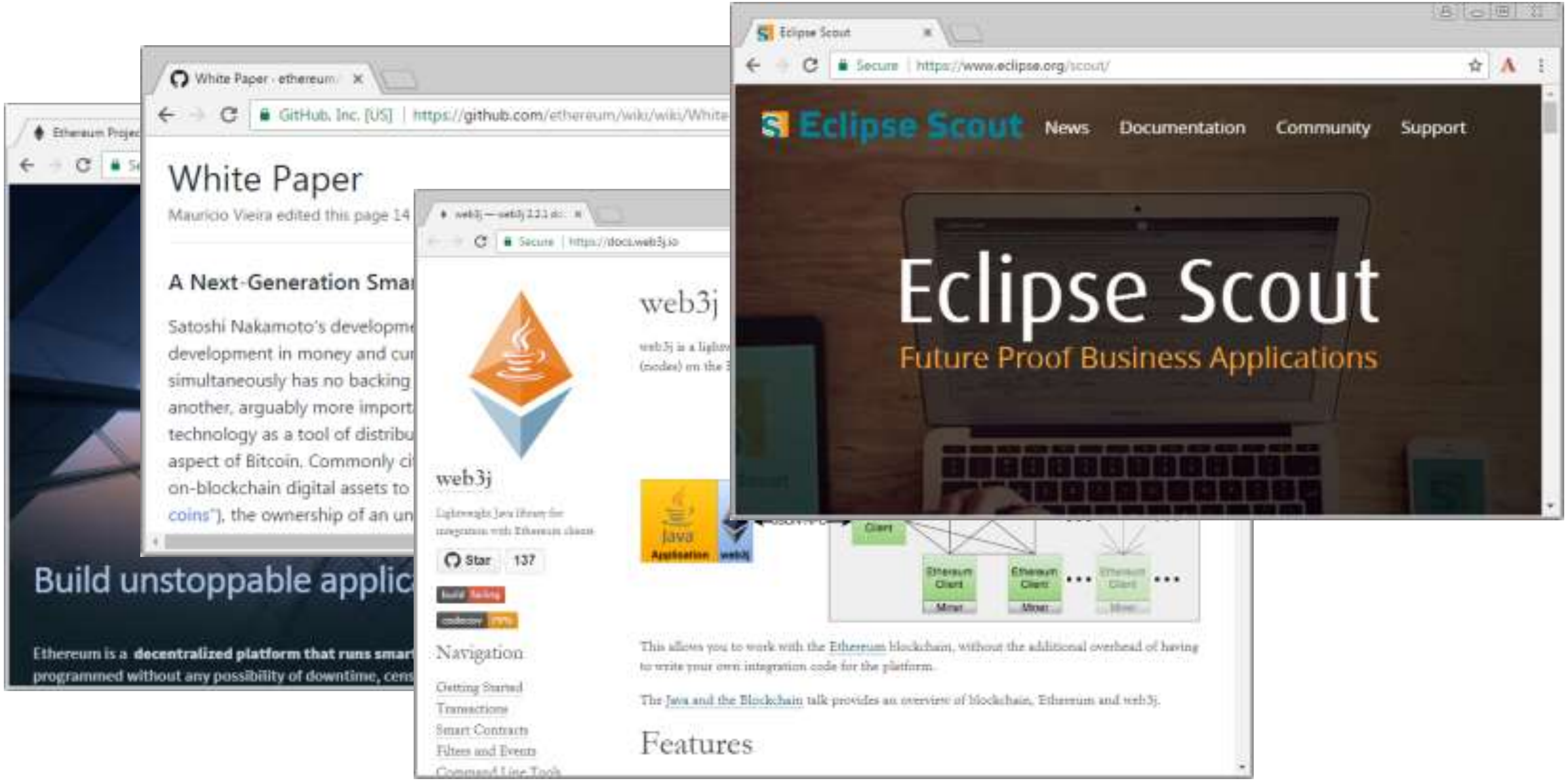


Deal-Nr.	Action	Organization	Quantity	Exchange rate	Quantity	Organization	Action
4				1.00	400,000	Roche	Sell
3				0.90	200,000	Roche	Sell
5				0.82	100,000	Roche	Sell
1	Buy	Nestlé	500,000	0.85			
2	Buy	Nestlé	1,000,000	0.80			

Sell	200,000	0.90	Published	USDEUR 3
Sell	400,000	1.00	Published	USDEUR 4
Sell	100,000	0.82	Pending	

```
> var USDEUR = OrderBoo
undefined
>
> USDEUR.symbol()
'USDEUR'
> USDEUR.getNumberOfSellOrders()
[ [String: '3'] s: 1, e: 0, c: [ 3 ] ]
> USDEUR.matchExists()
true
> USDEUR.topBuyOrderId()
[ [String: '1'] s: 1, e: 0, c: [ 1 ] ]
> USDEUR.topSellOrderId()
[ [String: '5'] s: 1, e: 0, c: [ 5 ] ]
```

Resources



Ethereum Projec

Build unstoppable applic

Ethereum is a **decentralized platform that runs smar**
programmed without any possibility of downtime, cens

White Paper · ethereum/

GitHub, Inc. [US] | https://github.com/ethereum/wiki/wiki/White

White Paper

Maurício Vieira edited this page 14

A Next-Generation Smar

Satoshi Nakamoto's developme
development in money and cur
simultaneously has no backing
another, arguably more import
technology as a tool of distribu
aspect of Bitcoin. Commonly ci
on-blockchain digital assets to
coins"), the ownership of an un

web3j — web3j 2.2.1 d

Secure | https://docs.web3j.io

web3j

Lightweight Java library for integration with Ethereum clients

Star 137

build failing

codecov 79%

Navigation

Getting Started
Transactions
Smart Contracts
Filters and Events
Command Line Tools

web3j

web3j is a lightw
(nodes) on the E

This allows you to work with the Ethereum blockchain, without the additional overhead of having to write your own integration code for the platform.

The Java and the Blockchain talk provides an overview of blockchain, Ethereum and web3j.

Features

Eclipse Scout

Secure | https://www.eclipse.org/scout/

Eclipse Scout

News
Documentation
Community
Support

Eclipse Scout

Future Proof Business Applications

What next?

Blockchain Summary

Gist

- ➔ Cool new technology, including much hype ☺
- ➔ Internet of decentralized trust

Blockchain Technology is great for

- ➔ Efficient value exchange for untrusted environments
- ➔ Pushes distributed business models
- ➔ Option for the «unbanked»

Current Challenges

- ➔ Privacy
- ➔ Scalability
- ➔ Maturity (blockchain still in ist infancy)

Next Steps

Socalizing

- ➔ Go to talks
- ➔ Join meetups (Bitcoin Meetup Switzerland, Blockchain Meetup Switzerland, Crypto Valley Forum, …)

Increase Context

- ➔ Youtube, Blogs, Twitter, …

Doing

- ➔ GitHub (web3j/web3j, matthiaszimmermann/web3j_demo, …)

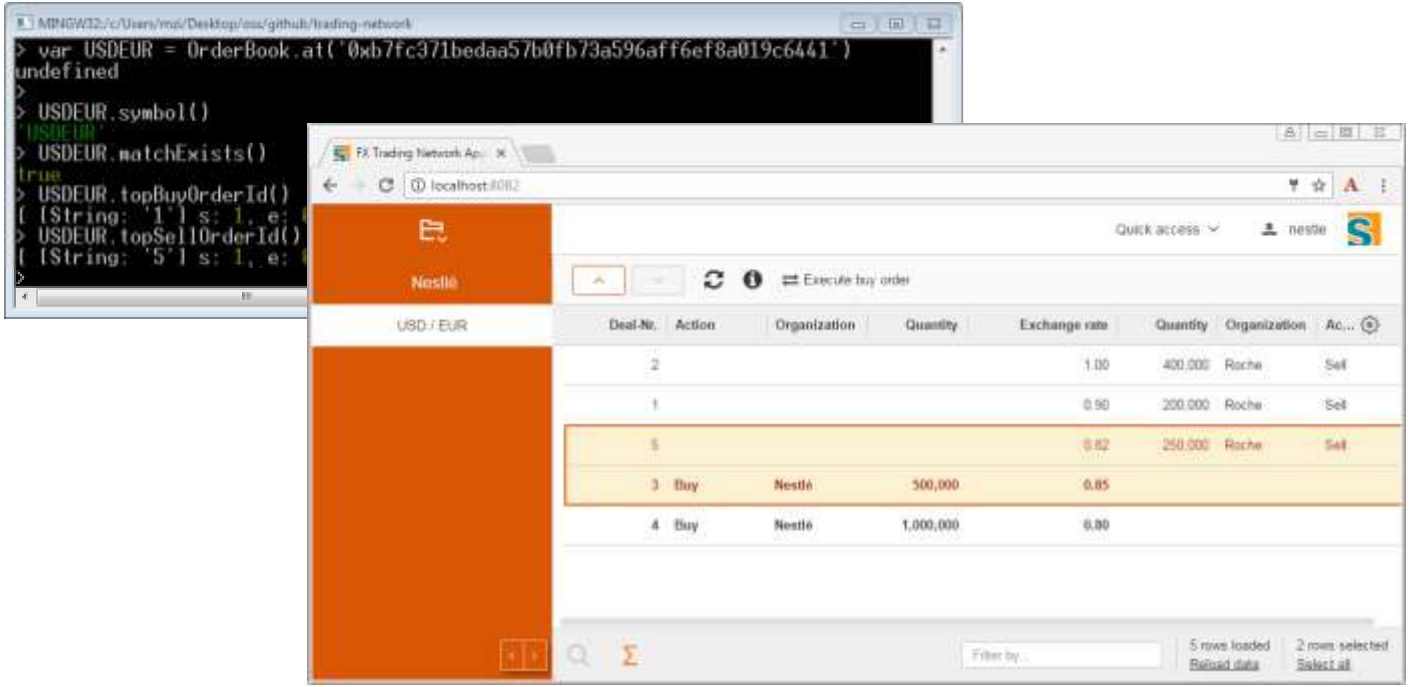
Everybody can do this ☺

«Recipe»

1. Curious about new technologies? Take yourself seriously!
2. Invest some of your time
3. Take advantage of your education options (time, money, …)
4. Building small teams makes it even more fun
5. Create value for your employer (internal, external)
6. Do it!

Thanks!

@ZimMatthias



```
> var USDEUR = OrderBook.at('0xb7fc371bedaa57b0fb73a596aff6ef8a019c6441')
undefined
>
> USDEUR.symbol()
'USDEUR'
> USDEUR.matchExists()
true
> USDEUR.topBuyOrderId()
[ [String: '1'] s: 1, e:
> USDEUR.topSellOrderId()
[ [String: '5'] s: 1, e:
>
```

Deal-Nr.	Action	Organization	Quantity	Exchange rate	Quantity	Organization	Ac...
2				1.00	400,000	Roche	Sell
1				0.90	200,000	Roche	Sell
5				0.82	250,000	Roche	Sell
3	Buy	Nestlé	500,000	0.85			
4	Buy	Nestlé	1,000,000	0.80			