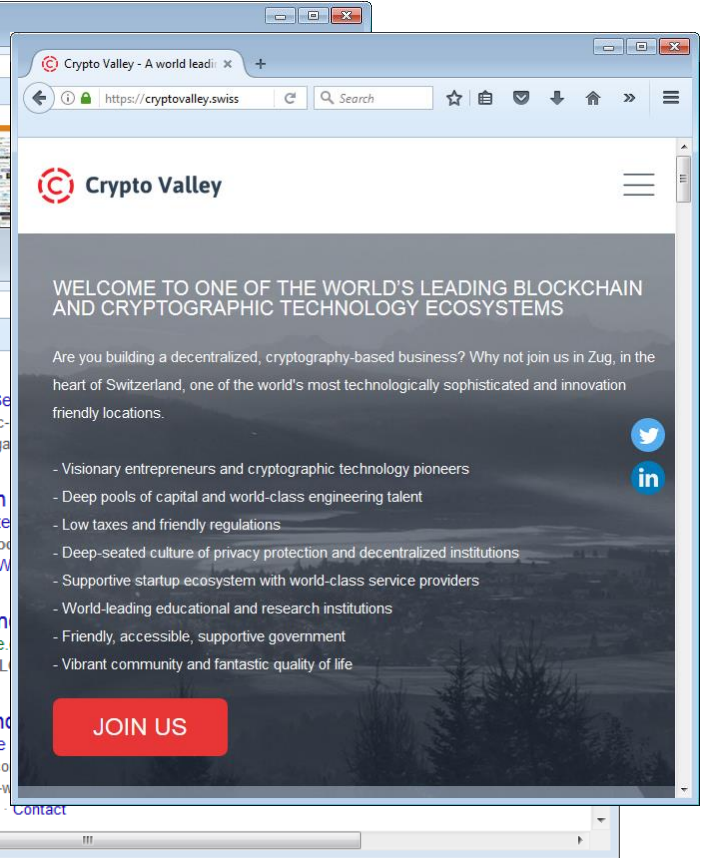
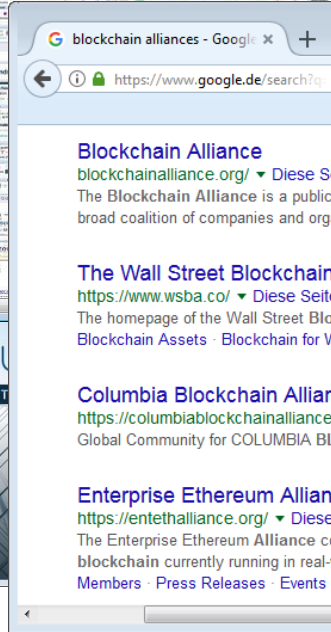
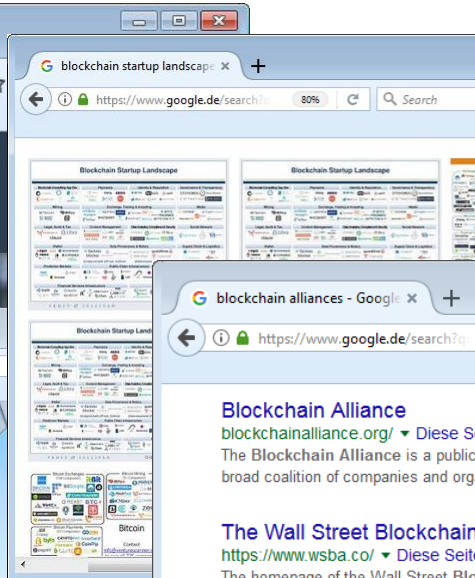
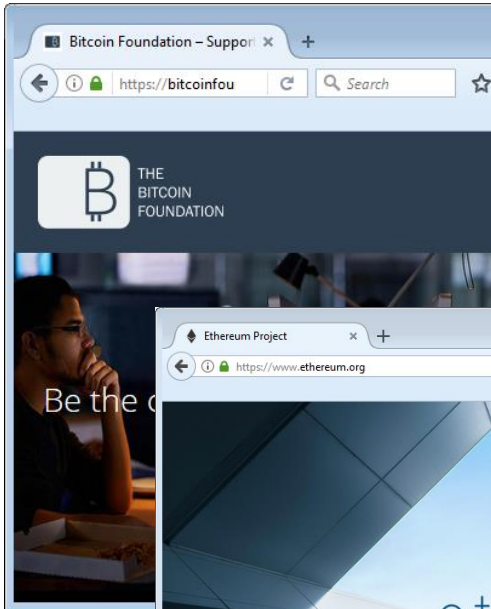
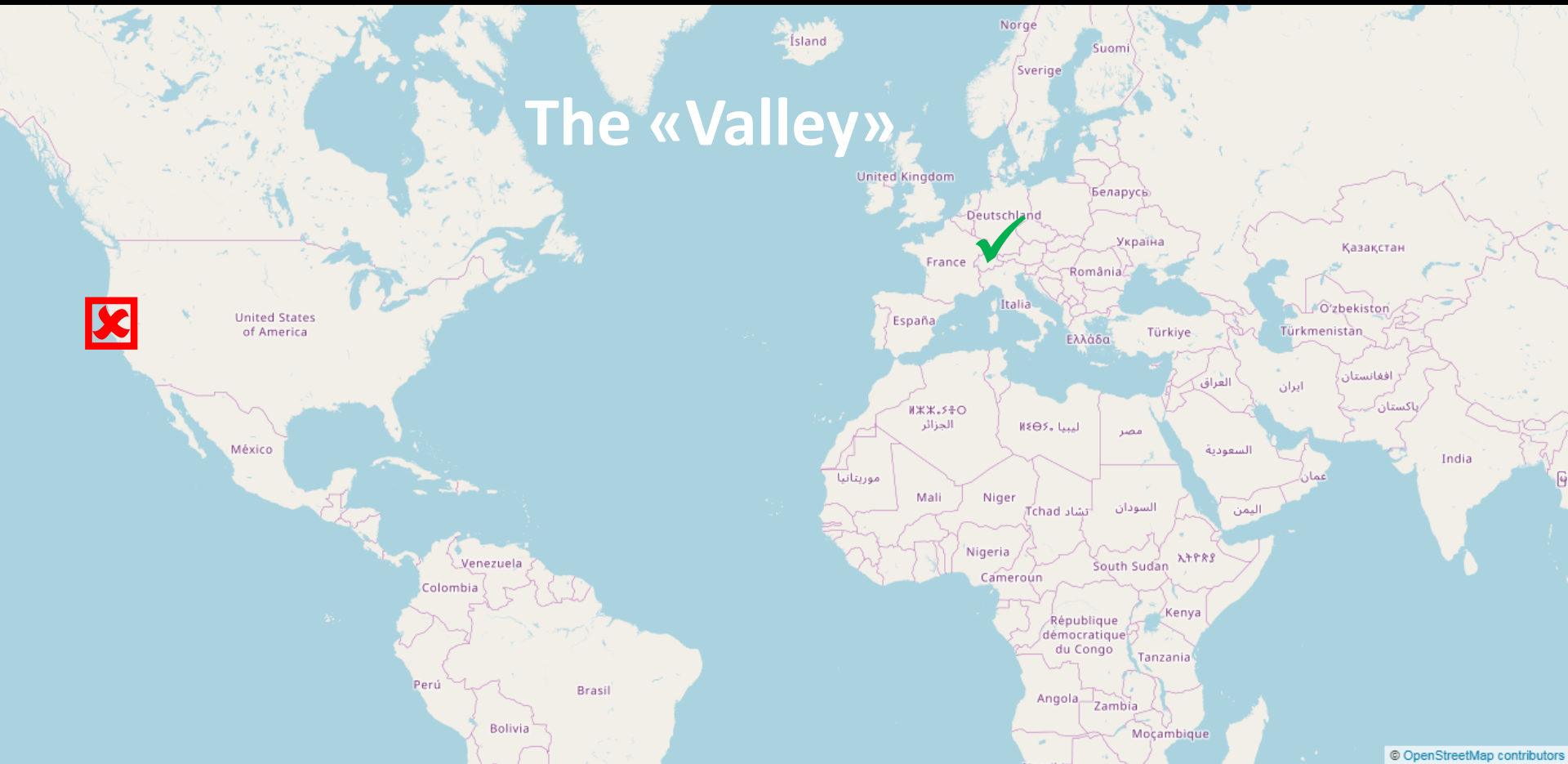


# Blockchain, Ethereum and Business Applications









@ZimMatthias Matthias Zimmermann  
BSI Business Systems Integration AG



# The «Valley»



# Blockchain Market

#	Name	Market Cap	Price	Volume (24h)	Circulating Supply
1	 Bitcoin	\$96,108,095,812	\$5779.73	\$1,929,760,000	16,628,475 BTC
2	 Ethereum	\$31,326,302,991	\$329.24	\$482,471,000	95,147,607 ETH
3	 Ripple	\$9,936,089,410	\$0.257869	\$866,836,000	38,531,538,922 XRP *
4	 Bitcoin Cash	\$5,864,609,962	\$351.12	\$346,000,000	16,702,438 BSH
5	 Litecoin	\$3,358,412,564	\$62.88	\$188,348,000	53,412,332 LTC
6	 Dash	\$2,307,274,645	\$302.54	<u>\$38,470,400</u>	7,626,321 DASH
7	 NEM	\$2,025,387,000	\$0.225043	\$8,065,810	8,999,999,999 XEM *
8	 Monero	\$1,451,708,799	\$95.32	\$31,838,500	15,229,142 XMR

List by the [International Monetary Fund](#) (Partial forecasted estimates for 2017)<sup>[5]</sup>

Rank ↕	Country ↕	GDP (millions of Int\$) ↕
	<i>World</i>	<b>126,687,917</b>
1	 China <sup>[n 1]</sup>	23,194,411
—	 European Union <sup>[n 2]</sup>	20,852,702
2	 United States	19,417,144
84	 Croatia	100,006
85	 Côte d'Ivoire	95,887
125	 Macedonia	31,924
126	 Cyprus	31,093

elizabeth stark gefällt das



**Neil Woodfine 聂尔** @nwoodfine · 9 Std.

Understanding bitcoin VS. when to buy and when to sell

[Original \(English\) übersetzen](#)



23

357

818



**Towards understanding Bitcoin ...**



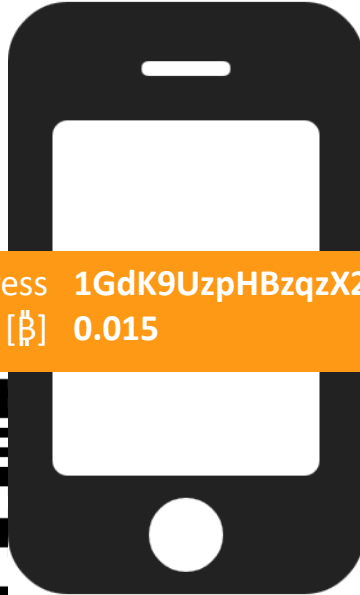
## «history's first» Blockchain

- Satoshi Nakamoto (?)
- White Paper 2008
- Open Source Software 2009

# Coffee at Bob's



Address **1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA**  
Amount [β] **0.015**





# What is a Bitcoin Address?

Addresses == «**Accounts**»

## Encoded Numbers

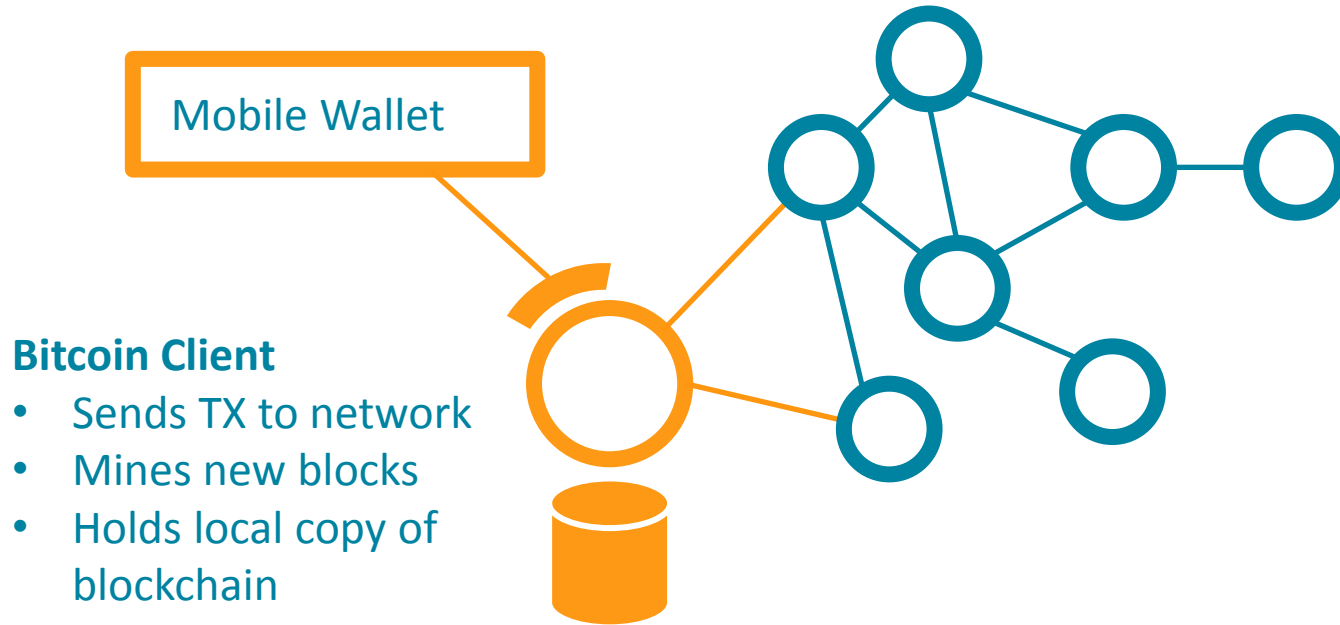
- Example: **1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA**
- Derived from a **public/private key pair**

## Getting, using and loosing it

- Create your own addresses, no need to ask anyone
- 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**

# Bitcoin Network

Peer-to-Peer Network



## Bitcoin Client

- Sends TX to network
- Mines new blocks
- Holds local copy of blockchain

# Distribution of unconfirmed TX

The screenshot shows a web browser window with the address bar displaying "Blockchain Luxembourg S.A.R.L. [LU] | https://blockchain.info/unconfirmed-transactions". The page content lists several unconfirmed transactions, each with a unique ID, a list of input addresses, a list of output addresses, and the amount in BTC. A green arrow points from the input addresses to the output addresses for each transaction. The browser window title is "198694 Unconfirmed Tra x".

Transaction ID	Input Addresses	Output Addresses	Amount (BTC)
eb78d5e6f397eb697d0dc4d360a63d4556bf503a40fa2ef3eed417eaa4a754df	1He94AHPRoSuPr1oSJ2cbY7RuRrYDvij6e 1KyubPL995GnaotonTA7eC5rnJ5jxEYyvg 1DXdroLk9EyhqQs1s9A5ZF5yFb7uMvaSo	1NymyGLqXFEvpDLux4bcP4NE9j6a3H8x8u 3GQKcb78cJL8n4nC9eJqKVBySaEwZYK	0.0509648 BTC 0.1515 BTC
184705ab578e28e232b744f778b431d0d2c5a32f91e25be0b8e1d67c21a5242c	199kQQSMfUWRaLJ7mVGN6SQjEHegC3ggD 16SdoLeciYdMTVQwsKqNmXVnxv7AmPjCbc 199kQQSMfUWRaLJ7mVGN6SQjEHegC3ggD	1DExtP6Wdgmz9cwEZqzYUCzMypqSmp7Xj 1K1ATEVpfpjRtFuMaMSZnT4cgnDAQT8DA9	2 BTC 0.553554 BTC
657738810e39b824c6367b23ac544ab1249257a3a4dee5d3d86300ed229942eb	14JksBEB3RbE39iqFpJ58LHBZ22Neu3iJ8	169W8GEPzPfdT9p2MeYRWP8bx4TEKqjND	0.05975 BTC

# The Coffee Transaction

- To confirm TX:
  - TX mined into block
  - 5 new blocks on top
- Network mines TX in ~ 10 min.
- Sometimes this takes hours ...

The screenshot shows a Bitcoin transaction page on blockchain.info. The URL is <https://blockchain.info/tx/0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8a>. The page displays the transaction ID, the sender's address (1CdId9KFAaatwczBwBttQcwXYCpvK8h7FK), and the receiver's address (1GdK9UzpHBzqzX2A9JFP3Di4weBwgmoQA). The transaction amount is 0.1015 BTC, and the fee is 0.0095 BTC. The transaction is confirmed 159578 times and included in block 277316.

Annotations on the screenshot:

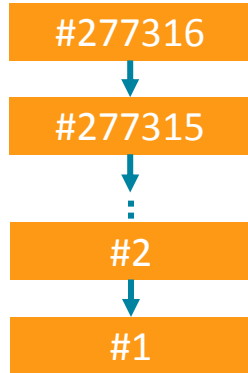
- Bob's Address**: Points to the receiver's address (1GdK9UzpHBzqzX2A9JFP3Di4weBwgmoQA).
- Coffee Price**: Points to the total output amount (0.1015 BTC).
- My Address**: Points to the sender's address (1CdId9KFAaatwczBwBttQcwXYCpvK8h7FK).
- Link to TX Block**: Points to the transaction ID (0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8a).
- TX Fee**: Points to the transaction fee (0.0095 BTC).

Summary		Inputs and Outputs	
Size	258 (bytes)	Total Input	0.1 BTC
Received Time	2013-12-27 23:03:05	Total Output	0.0995 BTC
Included In Blocks	277316 ( 2013-12-27 23:11:54 + 9 minutes )	Fees	0.0005 BTC
Confirmations	159578	Estimated BTC Transacted	0.015 BTC

Screenshot: blockchain.info

# TX Blocks and the Blockchain

→ **Block #277316**  
includes coffee TX



Block Hash  
do you see the leading zeros?

419 TX (€495,000)

### Block #277316

Summary	
Number Of Transactions	419
Output Total	10,322.07722534 BTC
Estimated Transaction Volume	777.75279147 BTC
Transaction Fees	0.09094928 BTC
Height	277316 (Main)
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	000000000000002a7bbd25a417c0374cc55261021e8a9ca74442b01284f0569
Next	00

Link to previous Block

Network Propagation (Click To View)

Screenshot: blockchain.info

# From Transactions to Blocks (Mining)

1. New TX are propagated through Bitcoin **peer-to-peer network**
2. Bitcoin client receive new TX and add 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**
  - 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

# Block Hashes and Crypto Challenge

```
mzi@BSI ~/Desktop/oss/events/2017/jug_blockchain_2017
```

```
$ java DummyBitcoinMiner 1
```

```
...
```

```
B37060F28617A5DFA3DB9A3D547663B3:I am Satoshi Nakamoto:lubf -> CF6BB3C636DF380E357E28271D948BBC
```

```
B37060F28617A5DFA3DB9A3D547663B3:I am Satoshi Nakamoto:jauj -> E6F1EB09AEF0BF331261B64E1C798EAB
```

```
B37060F28617A5DFA3DB9A3D547663B3:I am Satoshi Nakamoto:dwpt -> 0CEE0B104F133062073BDD810D3DB6CF
```

```
Success with nonce 'dwpt' for difficulty '0'. Hashes calculated: 29
```

```
$ java DummyBitcoinMiner 4
```

```
...
```

```
B37060F28617A5DFA3DB9A3D547663B3:I am Satoshi Nakamoto:quyc -> 89BD526A2DF7EBF41ACD227BFC6B78FE
```

```
B37060F28617A5DFA3DB9A3D547663B3:I am Satoshi Nakamoto:ceaw -> 0000AD30B2DDD8A773B8E5AABB5282AD
```

```
Success with nonce 'ceaw' for difficulty '0000'. Hashes calculated: 11584
```

Difficulty Oct' 17: 18 leading zeros

# Bitcoin Recap

## Bitcoin Success

- Completely decentral currency (no need for banks)
- Open Source (GitHub) and Open Data (complete TX history)
- First successful implementation of any crypto currency
- «Gold Standard» since 2009
- Record price levels in 2017

## Bitcoin Challenges

- Scaling debate/war
- Declining market share (Ethereum, ...)



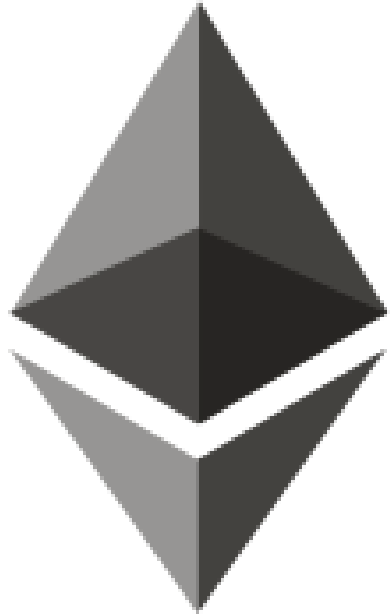
# Bitcoin Resources

The collage features three main elements:

- Book Cover:** The cover of 'Mastering Bitcoin: Unlocking Digital Cryptocurrency' by Andreas M. Antonopoulos, published by O'Reilly. It features a purple background with a detailed illustration of an ant carrying a leaf fragment.
- GitHub Repository:** A screenshot of the GitHub repository for 'bitcoin/bitcoin'. The page shows the repository name, navigation options (Code, Issues, Pull requests, Projects, Insights), and a commit history section. A recent commit by 'nadamsoreilly' is visible, along with a list of 22 contributors and file statistics (213 lines, 127 sloc, 33.3 KB).
- Document Snippet:** A snippet from a document titled 'Contributors to bitcoin'. It includes a bar chart showing contributions in 2010 and a section titled '1. Introduction' which discusses the nature of Bitcoin as a peer-to-peer network.

Andreas M. Antonopoulos

Ethereum



## Currency and Smart Contracts

- 2014 by Vitalik Buterin
- Distributed VM (Turing complete)
- Open source software

# Ethereum vs Bitcoin

## Common Traits

- Virtual currency
- Peer-to-peer network/nodes with local blockchain
- Concepts of addresses, transactions, mining

## Main Differences

- **Specification** with different implementations (Bitcoin reference client)
- **Smart contracts** and Ethereum virtual machine
- **Gas** to execute smart contracts and TX

# Gas

## What is gas?

- Unit to pay mining nodes
- Unit of gas has price in Ethers

## What can we buy with gas?

- Pay TX fees
- Execute smart contracts
- Computations performed by clients: **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)

# Ethereum Smart Contracts

## What is a Smart Contract?

- Code written in high level language «Solidity»
- Code is compiled into byte code
- Byte code executed on client by EVM
- Has owner
- Has address
- Can hold currency

## Examples

1. Flight delay insurance
2. «Truly» autonomous cars

# Flight Delay App

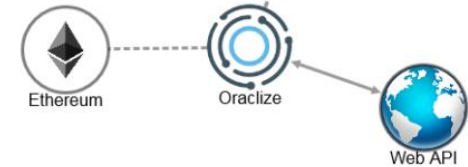
The screenshot shows the 'Your policy' page on <https://fdd.etherisc.com>. The page title is 'Your policy' with three numbered steps (1, 2, 3). Step 2 is active. The user is applying for a policy for flight AF650. The currency is set to EUR. The premium is 25€. There is a coupon code field. Below, a table shows payouts for premium: 25€.

Delay in minutes	15 - 29	30 - 44	45+	Cancelled	Diverted
Payout	54.83€	82.24€	164.48€	274.13€	274.13€

Buttons for 'Report an issue', 'Apply', and 'Apply' are visible.

The screenshot shows the flight status page for (AF) Air France 650 (CDG Paris, FR to (CUN) Cancun, MX) on <https://www.flightstats.com>. The status is 'Landed - Delayed 51 minutes'. The last change to status was more than 3 hours ago.

DEPARTURE	ARRIVAL
Scheduled Departure: 12:15 PM - Sun Oct-22-2017	Scheduled Arrival: 4:05 PM - Sun Oct-22-2017
Actual Departure: 1:27 PM - Sun Oct-22-2017	Actual Arrival: 4:56 PM - Sun Oct-22-2017
Arrival Terminal: 3	Baggage Claim: N/A



# «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

f SHARE t TWEET in LINKEDIN



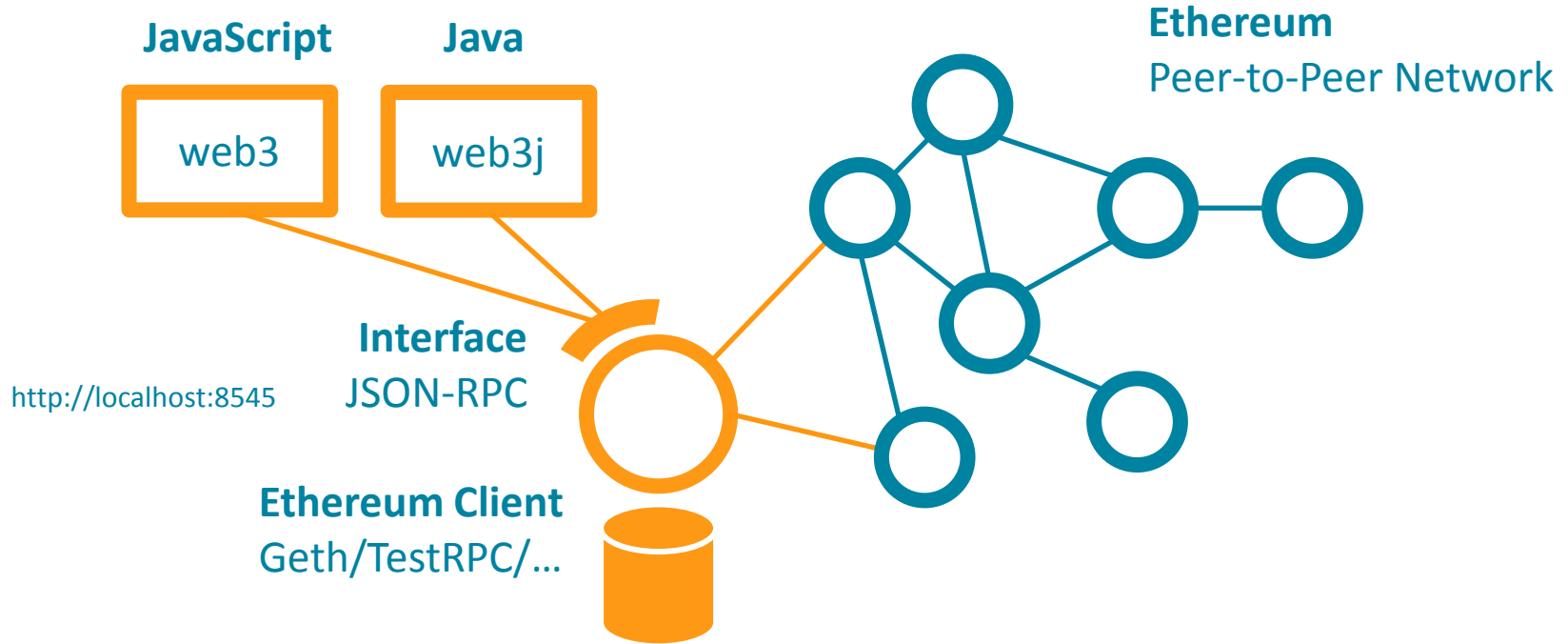
In 2017 RWE introduces charging stations connected to Ethereum blockchain

- **Smart contract:** Order car to transport people (fees go to car contract)
- **Smart contract:** Car pays for energy/services (car pays with its own funds)



«Greeter» with web3  
Smart Contract Hello World

# Ethereum and Applications



# «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);  
    }  
}
```

```

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() { ... }
}

```

## greeter.sol

+ additional state

+ 'payable'

# Solidity Compiler (online)

The screenshot displays the online Solidity compiler interface. On the left, the source code for a contract named 'greeter' is shown. The code includes a pragma statement, a constructor, a payable function, and a main 'greet' function. An orange callout box points to the 'Bytecode' section, which displays the hex string '6060604052341561000c57fe5b604051610407380380610'. Another orange callout box points to the 'Web3 deploy' section, which contains a JavaScript deployment script. The script defines a contract instance and a function to mine the contract.

```
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
```

```
    }
```

```
}
```

byte code (EVM)  
to deploy contract

Bytecode

```
6060604052341561000c57fe5b604051610407380380610
```

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: '0x6060604052341561000c57fe5b6040516
```

```
    gas: '4700000'
```

```
}, function (e, contract){
```

```
    console.log(e, contract);
```

```
    if (typeof contract.address !== 'undefined'
```

```
        console.log('contract mined! address:
```

```
    }
```

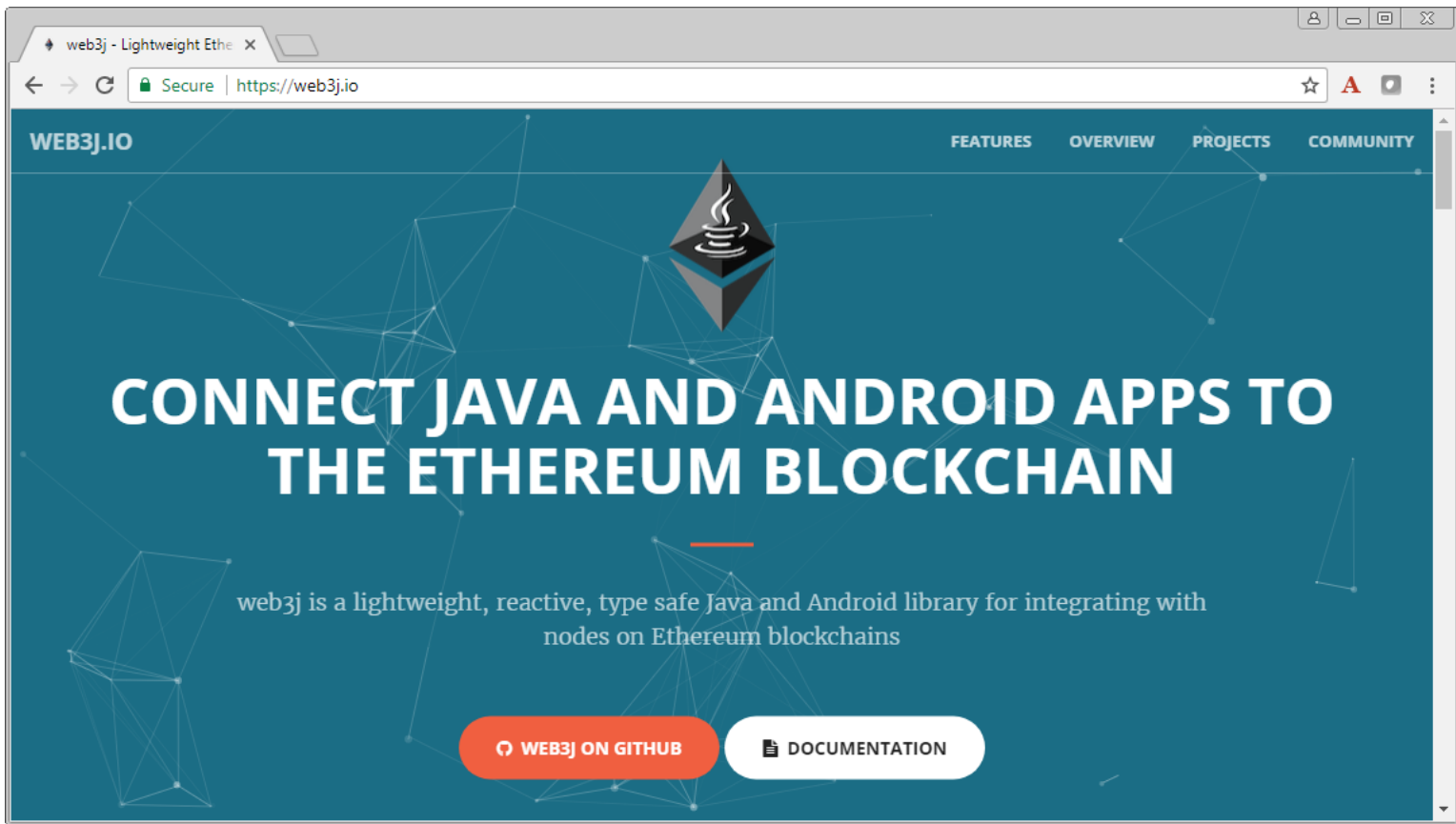
```
})
```

Deploy script (JS)

# Deploy (Console)

```
MINGW32:/c/Users/mzi/Desktop/private/github/web3j_demo
greet:
{ [Function: bound ]
  request: [Function: bound ],
  [Function: bound ],
  [Function: bound ],
  getData: [Function: bound ],
  [Circular] },
allEvents: [Function: bound ] }
Contract mined! address: 0xa0bcf163159ef958dc360e1dafdcd1dc8af3939d transactionHash: 0xc317df210
> greeter.greet()
'hello world!'
>
> undefined_greeter.deposits();
{ [String: '0'] s: 1, e: 0, c: [ 0 ] }
>
```

**«Greeter» with web3j**



# CONNECT JAVA AND ANDROID APPS TO THE ETHEREUM BLOCKCHAIN

web3j is a lightweight, reactive, type safe Java and Android library for integrating with nodes on Ethereum blockchains

[WEB3J ON GITHUB](#)

[DOCUMENTATION](#)



# greeter.sol → Greeter.java

1. Compile `greeter.sol`
  - `greeter.bin`
  - `greeter.abi`
2. Create wrapper class (use Web3j command line tool)
  - `Greeter.java`

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure for 'web3j\_demo', including source files like 'ContractDemo.java' and 'greeter.sol'. The main editor shows the code for 'ContractDemo.java', which includes a 'callGreet' method that interacts with a Greeter contract. The Console window at the bottom shows the output of a Java application, detailing the deployment of the Greeter contract, the funding process, the execution of the 'greet()' method, and the final state of the contract and Alice's account balance.

```
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
Package Explorer
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[]): void
    deployContract(): Greeter
    sendFunds(Greeter): void
    callGreet(Greeter): void
    killContract(Greeter): void
    fundAlice(): void
    printBalanceAlice(String): void
ContractDemo.java
greeter.sol
99
100 private void callGreet(Greeter contract) throws Exception {
101     System.out.println("// Call greet()");
102
103     Utf8String message = contract
104         .greet()
105         .get();
106
107     System.out.println("Message returned by Contract.greet(): " + message.toString());
108     printBalanceAlice("after greet");
109     System.out.println();
110 }
Problems @ Javadoc Declaration Console Tasks
<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
Writable Smart Insert 100 : 64
```

# Trading-Network Demo

Ethereum, web3j, Eclipse Scout

# Trading Network Demo

## Use Case

- **Currency Hedging**: Manage buy and sell orders for € / US\$
- **Classical Business App**
  - **Identity management**: Map users with Ethereum addresses
  - **web3j** library to access Ethereum client
  - **Eclipse Scout** to build application
- **Blockchain Benefits**
  - **Efficiency**: No central organization/infrastructure
  - **Trust**: Tampering-proof ledger, trust by blockchain

 **Eclipse Scout**  
UI (web application)




 **Eclipse Scout**  
Backend


**web3j**      **JDBC**

 **web3**

**Ethereum Client**  
TestRPC

 **PostgreSQL**

FX Trading Network App | X  
localhost:8082

Quick access | nestle | 


Welcome to your FX-Trading Tool

Country: Switzerland


**Nestlé**


- Own Deals
- USD / EUR
- > Trading Center

**Balance**


Currency	Balance	Refresh	


**Overview**


Information	Count	
USD / EUR - Buy - Inactive	2	

Quick access | roche | 

**Roche**

Information	Count	
USD / EUR - Sell - Inactive	2	

Currency	Balance	Refresh	

Information	Count	
USD / EUR - Sell - Inactive	2	

Nestlé

Own Deals

USD / EUR

Trading Center

Execute buy order

Deal-Nr.	Action	Organization	Quantity
4			
3			
5			
1	Buy	Nestlé	500,000
2	Buy	Nestlé	1,000,000

```
MINGW32: c:/Users/mzi/Desktop/oss/github/trading-network  
> var USDEUR = OrderBook.at('0xb7fc371bedaa57b0fb73a596aff6ef8a019c6441')  
undefined  
>  
> USDEUR.symbol()  
'USDEUR'  
> USDEUR.getNumberOfBuyOrders()  
rOfSellOrders()  
USDEUR.matchExists()  
USDEUR.topBuyOrderId() { [String: '2'] s: 1, e: 0, c: [ 2 ] }  
> 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 ] }  
> USDEUR.matchExists()  
false  
>
```

Trading Center

USD / EUR	Sell	100,000	0.82	Pending
-----------	------	---------	------	---------

Filter by...  
3 rows loaded  
Reload data  
One row selected  
Select all

# Resources

The image displays four overlapping browser windows showcasing resources related to Ethereum and web3j:

- White Paper - ethereum/**: A GitHub page for the Ethereum White Paper, edited by Mauricio Vieira. The text includes: "A Next-Generation Smart Contract Platform", "Satoshi Nakamoto's development in money and currency simultaneously has no backing other than trust in the system, another, arguably more important aspect of Bitcoin. Commonly cited as a tool of distributed ledger technology as a tool of distributed ledger technology (often referred to as 'digital assets' or 'coins'), the ownership of an asset is recorded on a distributed ledger.", and "Build unstoppable applications". Below this, it states: "Ethereum is a **decentralized platform that runs smart contracts** programmed without any possibility of downtime, censorship, fraud or third-party interference."
- web3j**: A GitHub repository page for the web3j library. It features the web3j logo (a blue diamond with a white coffee cup) and the text: "web3j is a lightweight Java library for integration with Ethereum clients". It shows 137 stars and a build status of "failing".
- Eclipse Scout**: The website for Eclipse Scout, featuring a dark background with a laptop and the text: "Eclipse Scout Future Proof Business Applications". The navigation menu includes: News, Documentation, Community, Support.
- web3j**: A page from docs.web3j.io, featuring the web3j logo and the text: "web3j is a lightweight Java library for integration with Ethereum clients". It includes a "Navigation" section with links to "Getting Started", "Transactions", "Smart Contracts", "Filters and Events", and "Command Line Tools". It also has a "Features" section with the text: "This allows you to work with the [Ethereum](#) blockchain, without the additional overhead of having to write your own integration code for the platform." and "The [Java and the Blockchain](#) talk provides an overview of blockchain, Ethereum and web3j."

<https://www.ethereum.org/> <https://github.com/ethereum/> <https://docs.web3j.io/> <https://www.eclipse.org/scout/>



# Summary

# Wrap-up

## Blockchain

- Internet of decentralized trust
- Cool new technology

## Technology is great for

- «unbanked»: Machines AND people
- Efficient trustless global exchange for values (and information)
- Distributed business models

## Challenges

- Privacy
- Scalability (Bitcoin can do 7 tx/s)
- Energy consumption (Bitcoin power consumption ~ Ecuador)
- Maturity (blockchain still in it's infancy)
- Regulatory (it's a mess ...)

# Thanks!

«Blockchain, Ethereum and Business Applications»



Evaluate the Sessions

Sign in and vote at [eclipsecon.org](https://eclipsecon.org)

- 1      0      + 1

**Additional Material**

Bitcoin · \$5,554.90    Ethereum · \$298.16    Litecoin · \$55.19

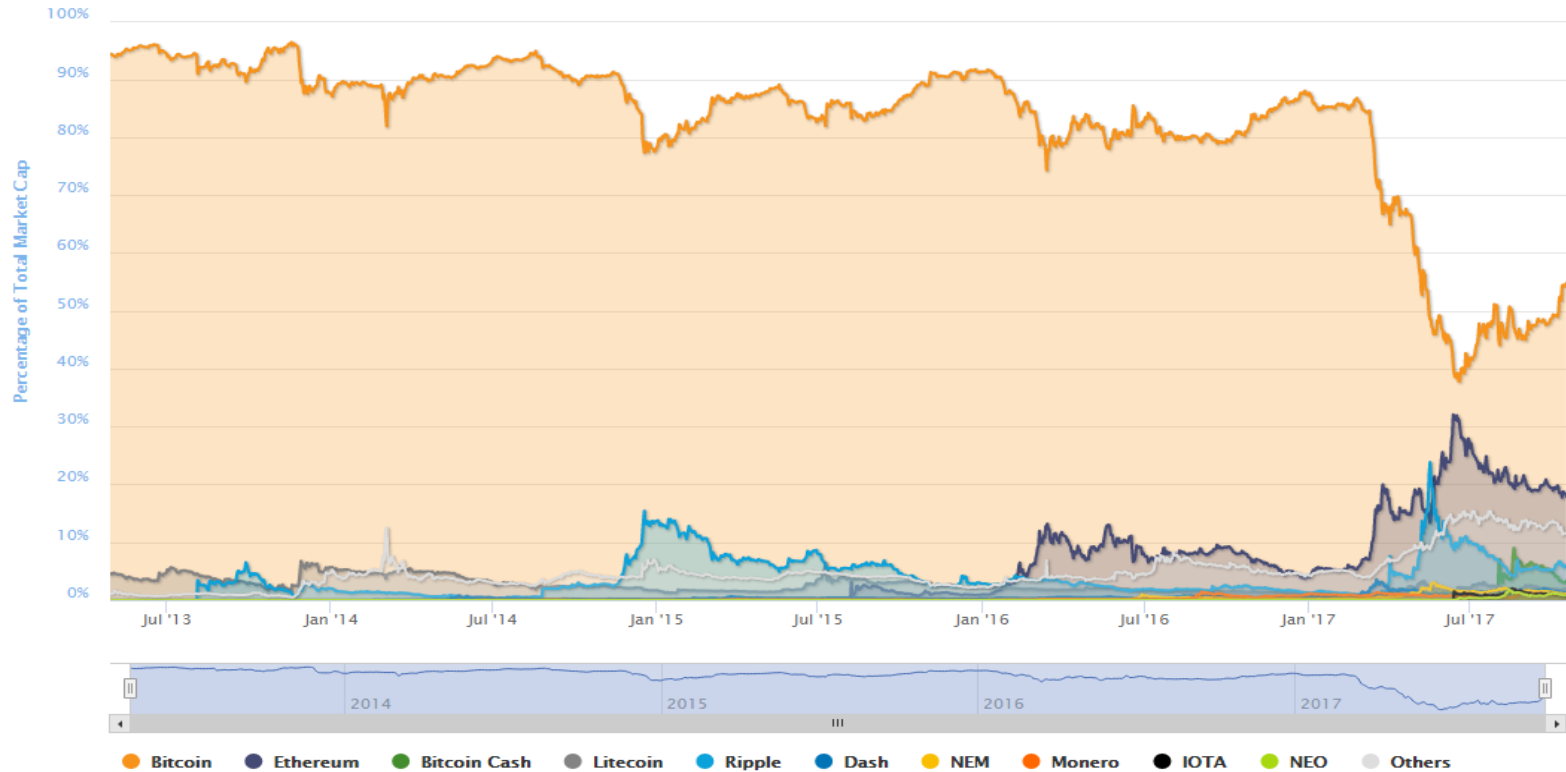
1H 1D 1W 1M 1Y ALL

# \$5,554.90

BITCOIN PRICE



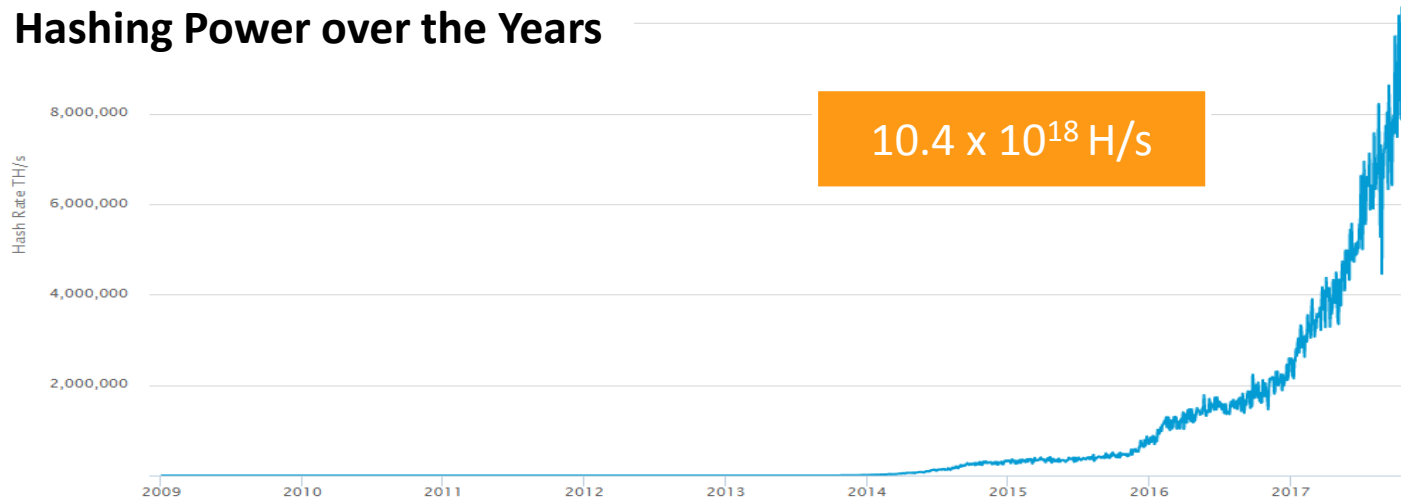
# Bitcoin Market Share



# 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(!)

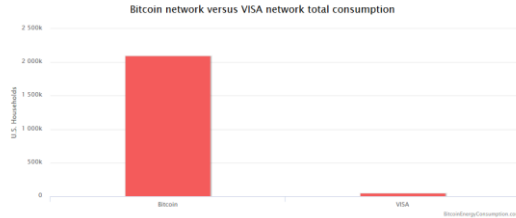
## Hashing Power over the Years



# Bitcoin Energy Consumption Index

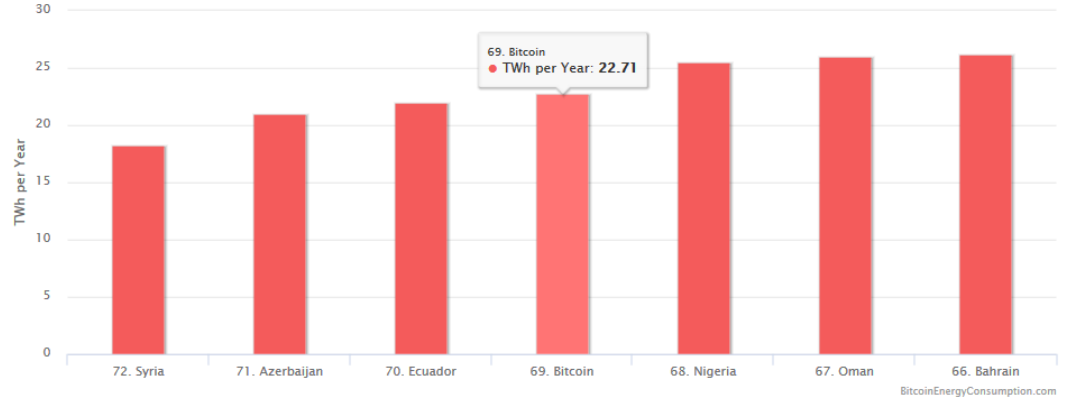
## Comparing Bitcoin's energy consumption to other payment systems

To put the energy consumed by the Bitcoin network into perspective we can compare it to another payment system like VISA for example. Even though the available information on VISA's energy consumption is limited, we can establish that the data centers that process VISA's transactions consume energy equal to that of 50,000 U.S. households. We also know VISA processed 82.3 billion transactions in 2016. With the help of these numbers, it is possible to compare both networks and show that Bitcoin is extremely more energy intensive per transaction than VISA.



Bitcoin vs VISA

## Energy Consumption by Country Chart



Bitcoin compared to countries



# Consensus Mechanism

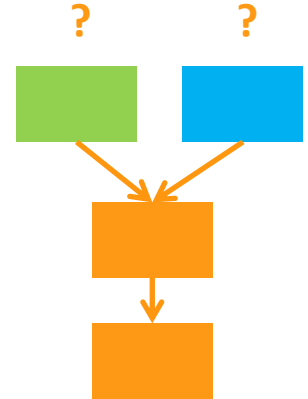
## «Preventing Forks»

### The Challenge

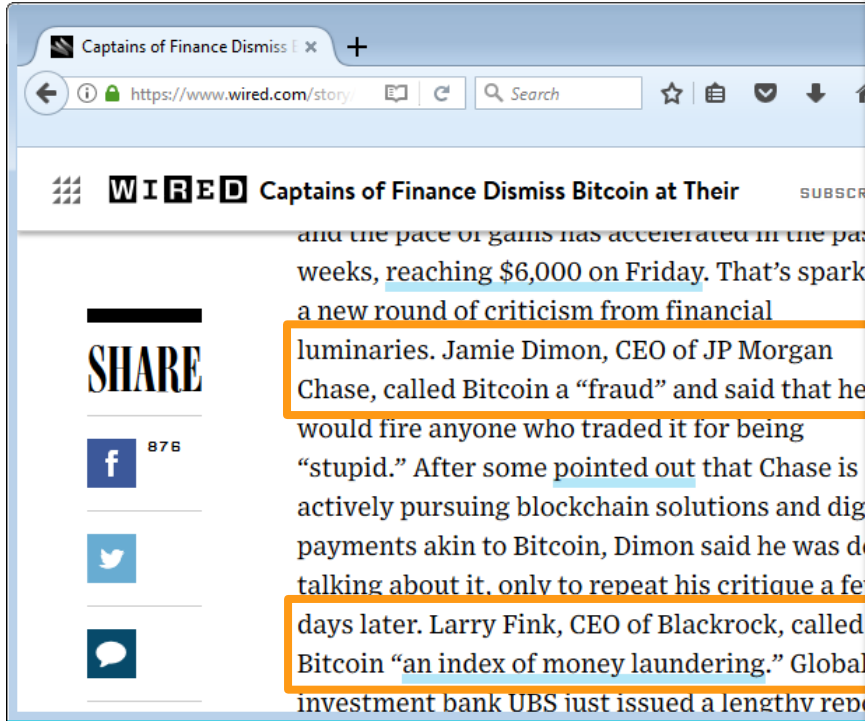
- Mining clients build block candidates independently
- **Several new blocks** might be found **at the «same» time**
- Clients may receive new blocks that are inconsistent
- The local copy of the blockchain may have forks

### 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



# Financial Industry getting ne



Captains of Finance Dismiss Bitcoin at Their

and the pace of gains has accelerated in the past weeks, reaching \$6,000 on Friday. That's sparked a new round of criticism from financial luminaries. Jamie Dimon, CEO of JP Morgan Chase, called Bitcoin a "fraud" and said that he would fire anyone who traded it for being "stupid." After some pointed out that Chase is actively pursuing blockchain solutions and digital payments akin to Bitcoin, Dimon said he was doing talking about it, only to repeat his critique a few days later. Larry Fink, CEO of Blackrock, called Bitcoin "an index of money laundering." Global investment bank UBS just issued a lengthy report

**SHARE**

f 876

Twitter

Message



**CoinDesk** @coindesk Folge ich

'Massive Disruption': IMF's Lagarde Says Cryptocurrencies Must Be Taken Seriously [bit.ly/2kJnyga](https://bit.ly/2kJnyga)

Original (English) übersetzen



05:31 - 13. Okt. 2017

339 Retweets 427 „Gefällt mir“-Angaben

🌐 📺 📱 📄 📧 📧 📧 📧 📧 📧