2019 NORTH AMERICA CACS™
AN ISACA EVENT

Blockchain, Blockchain Security and the Basics of Blockchain Auditing

May 11 - 12, 2019

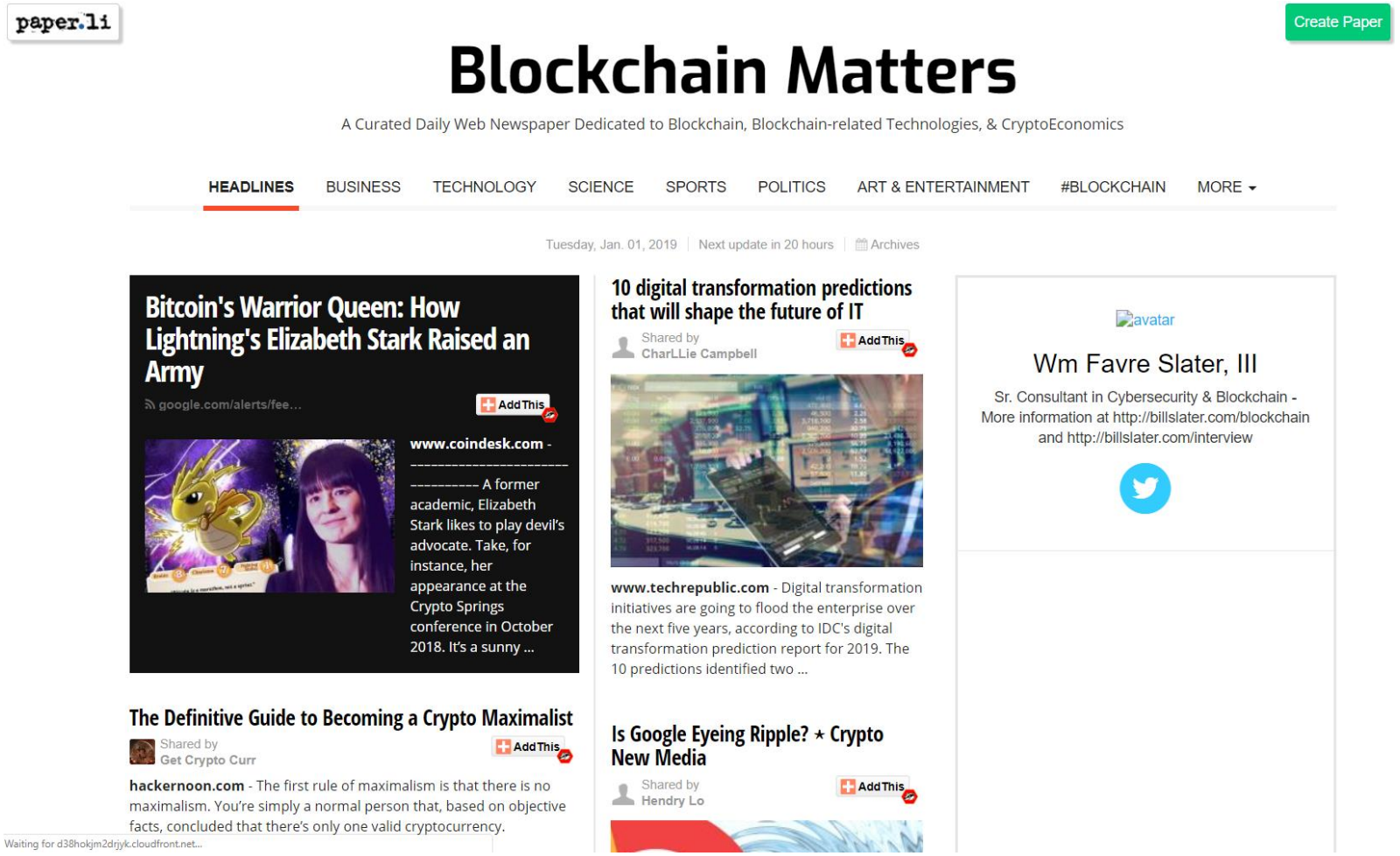
Day 1

William Favre Slater, III

M.S., MBA, PMP, CISSP, CISA, SSCP, Security+, ITILv3

ISACA Gold Member

Free Blockchain Daily Newspaper



More information: https://paper.li/billslater/1530793250#/

Agenda – Day 1 & Day 2

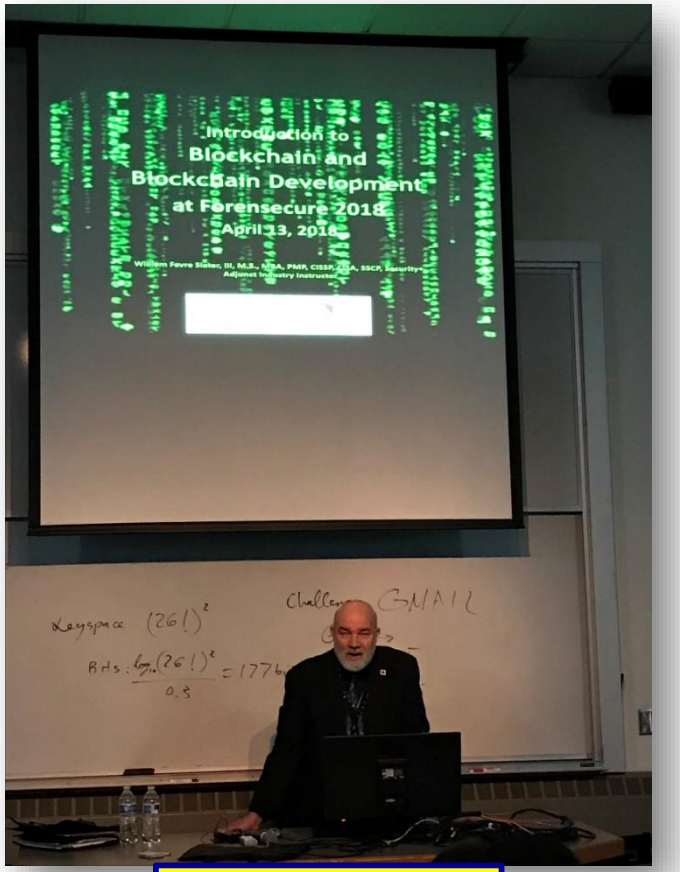
High-level Outline:

Day 1

Topic 1: History of Money and Conventional Ledger Functions
Topic 2: Bitcoin Basics
Topic 3: Tokenized Economy and Crypto Currency Concepts
Topic 4: Blockchain Technology
Topic 5: Ethereum Blockchain Technology
Topic 6: Blockchain Beyond Bitcoin
Topic 7: Blockchain Limits and Challenges
Topic 8: Blockchain Security
Topic 9: Examples of Real-world Blockchain Applications
Topic 10: The Ethereum EVM, Smart Contracts, and Solidity
Topic 11: How to Design and Implement a Blockchain Solution Project – an Organized High-Level Step-by-Step Approach
Topic 12: How to Help your Organization Rapidly Ramp Up Skills and Readiness for Blockchain Application Development

Day 2

Topic 1: Getting started with Blockchain Application Development – Setting up the Workbench
Topic 2: Truffle Framework Introduction
Topic 3: Example DApp using Truffle, HTML, CSS, Solidity, the EVM and Ethereum Blockchain
Topic 4: Solidity and Ethereum Blockchain Fundamentals
Topic 5: Javascript and Ethereum Blockchain Fundamentals
Topic 6: Example DApp using HTML, CSS, Solidity the EVM and the Ethereum Blockchain
Topic 7: Blockchain and Auditing
Topic 8: How to Secure Blockchain infrastructure and applications
Topic 9: How to perform Secure Software Development for Blockchain applications by design, coding practices, testing and verification
Topic 10: Concepts of Auditing the Data and Transactions in Blockchain Data Structures
Topic 11: Automating the Auditing of Blockchains and Blockchain Applications

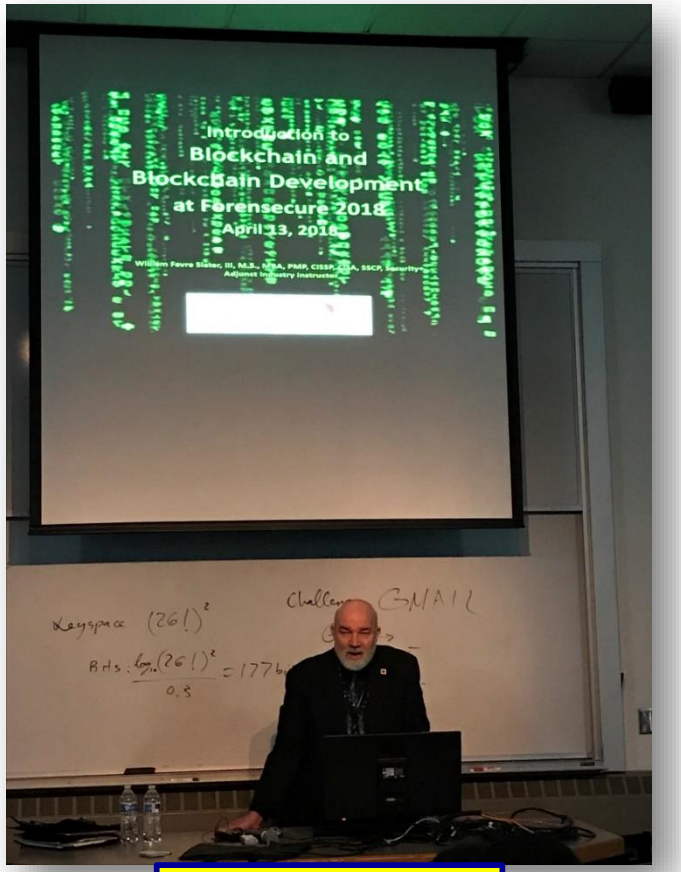


William Favre Slater, III
Forensecure 2018

Agenda – Day 1

Day 1

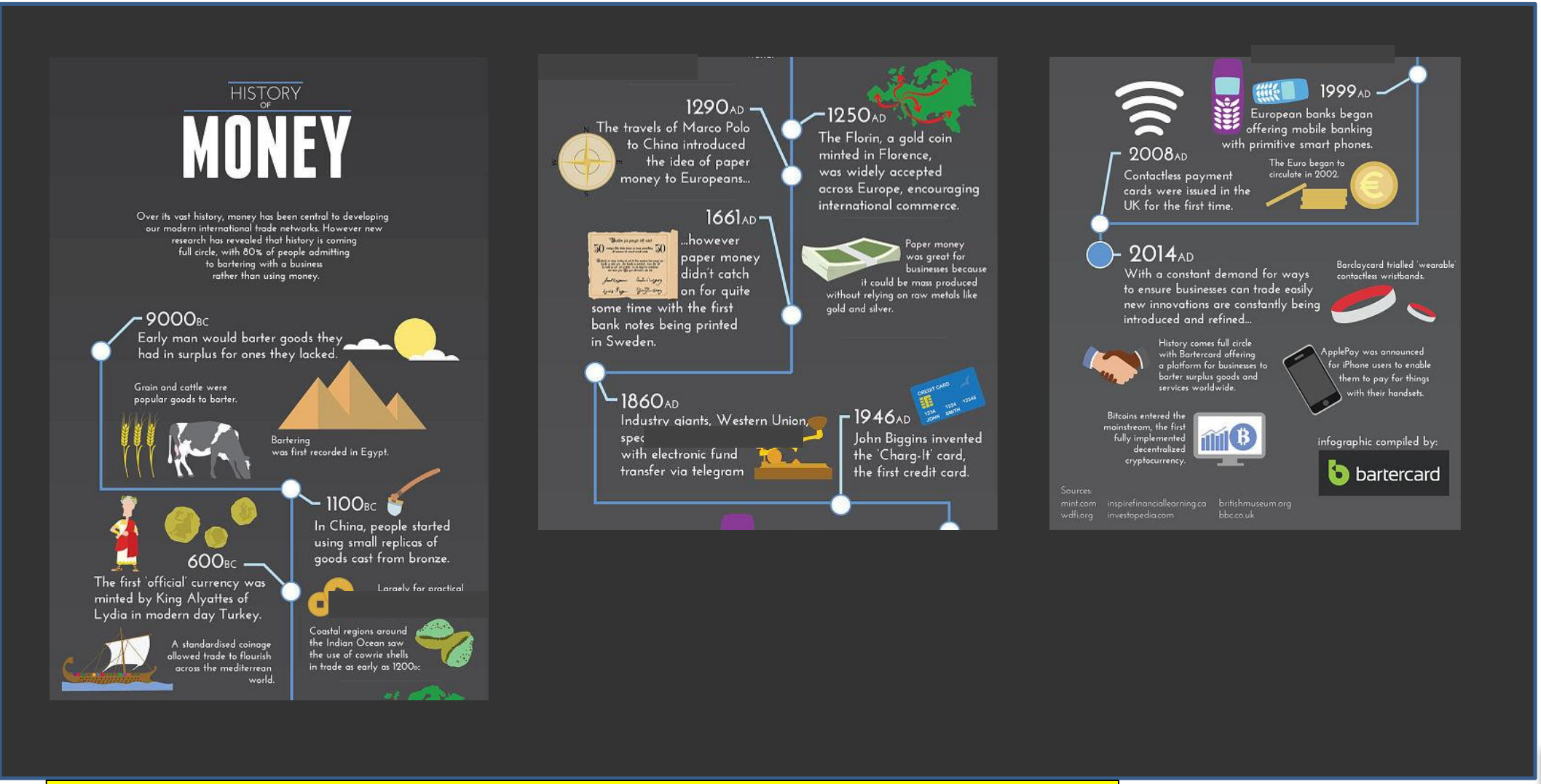
Topic 1: History of Money and Conventional Ledger Functions
Topic 2: Bitcoin Basics
Topic 3: Tokenized Economy and Crypto Currency Concepts
Topic 4: Blockchain Technology
Topic 5: Ethereum Blockchain Technology
Topic 6: Blockchain Beyond Bitcoin
Topic 7: Blockchain Limits and Challenges
Topic 8: Blockchain Security
Topic 9: Examples of Real-world Blockchain Applications
Topic 10: The Ethereum EVM, Smart Contracts, and Solidity
Topic 11: How to Design and Implement a Blockchain Solution Project – an Organized High-Level Step-by-Step Approach
Topic 12: How to Help your Organization Rapidly Ramp Up Skills and Readiness for Blockchain Application Development



William Favre Slater, III
Forensecure 2018

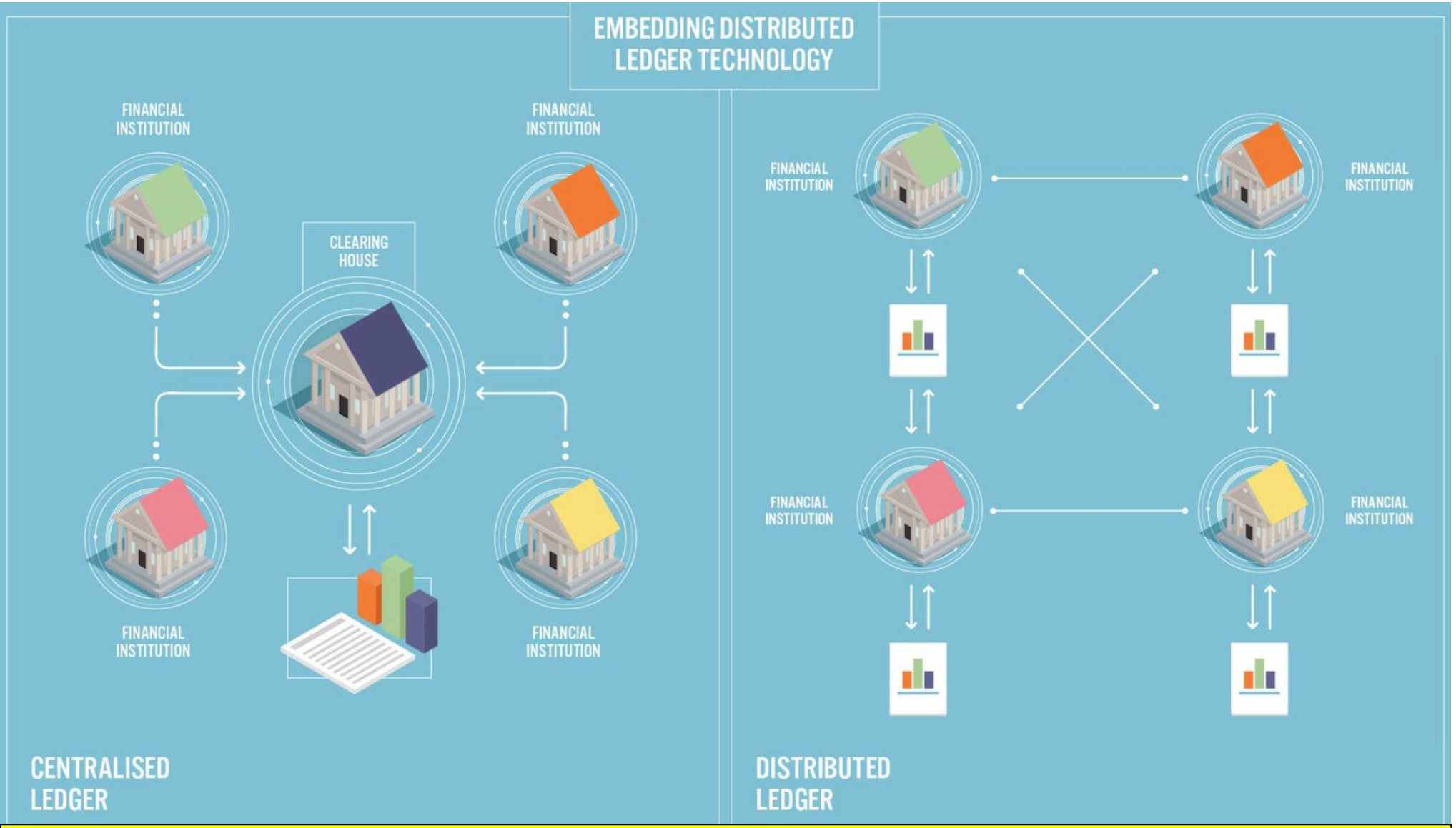
Topic 1: History of Money and Conventional Ledger Functions

Brief History of Money



Source: https://www.telegraph.co.uk/finance/businessclub/money/11174013/The-history-of-money-from-barter-to-bitcoin.html

Conventional vs. Distributed Ledger

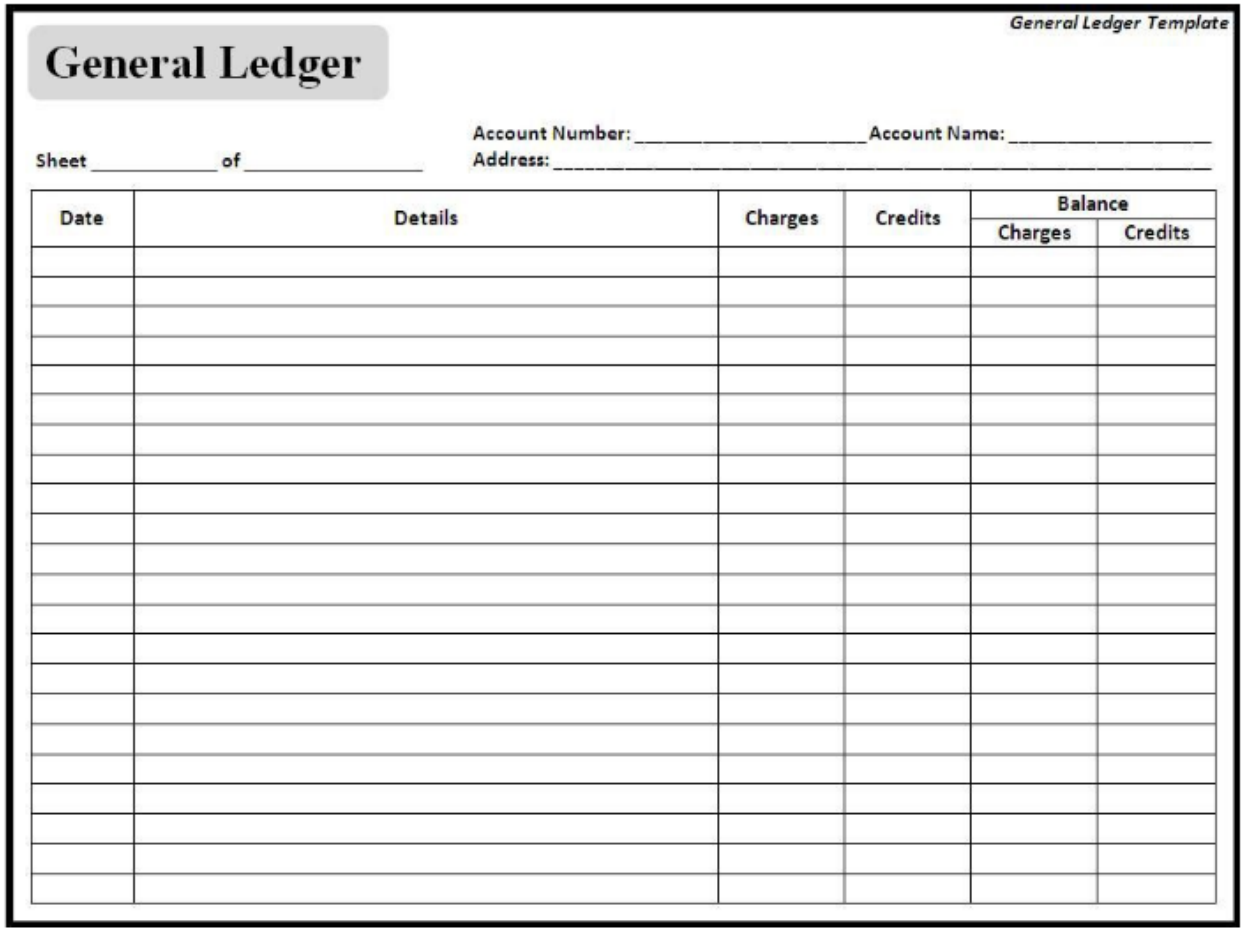


Source: https://codeburst.io/distributed-ledger-technology-fundamentals-you-must-know-2d0f82628258

History of Conventional Ledger Functions

The biggest problems with our existing accounting ledgers are:

- ***Inefficient to manage***
- ***Resource Intensive and expensive***
- ***Duplicate Entry Pops Up***
- ***Can be manipulated***
- ***Can be hacked***
- ***Lack of transparency***
- ***Data corruption, fraud, leading to disputes***



General Ledger Template

General Ledger

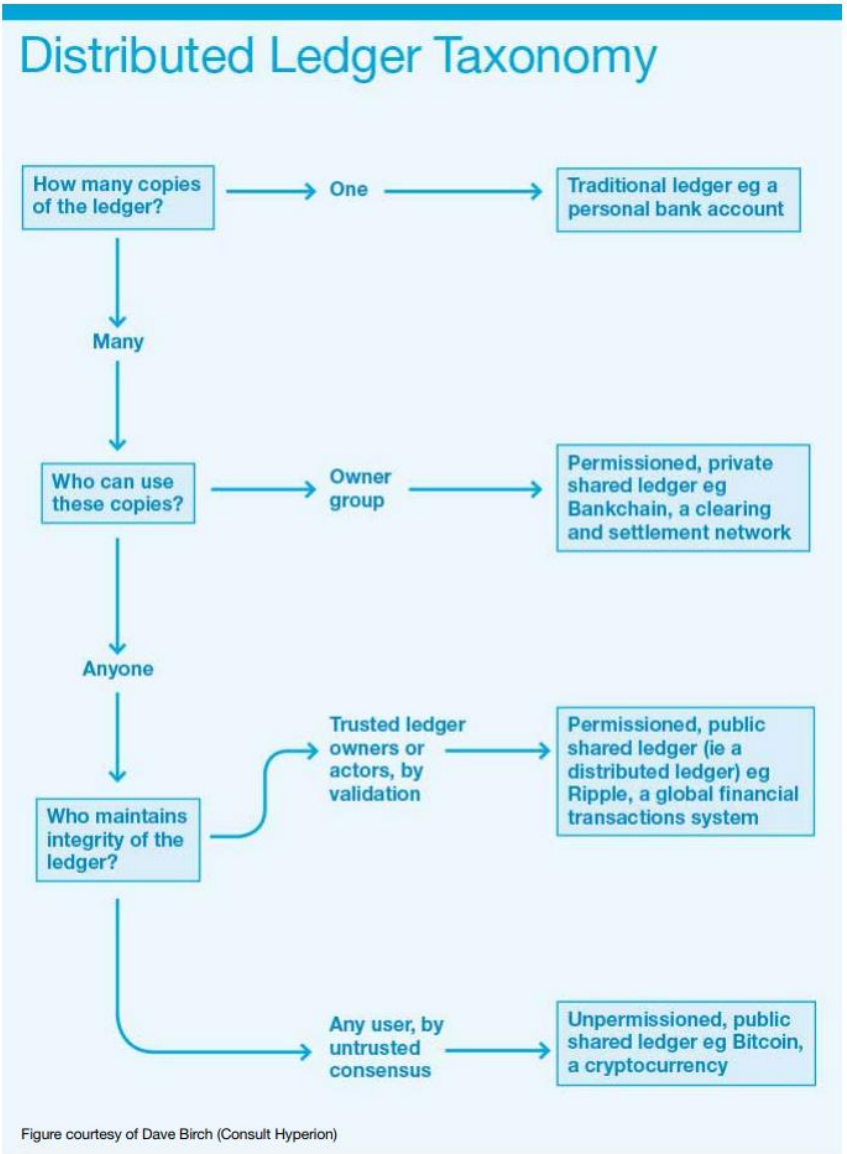
Account Number: _____ Account Name: _____

Sheet _____ of _____ Address: _____

Date	Details	Charges	Credits	Balance	
				Charges	Credits

Source: https://codeburst.io/distributed-ledger-technology-fundamentals-you-must-know-2d0f82628258

Distributed Ledger Taxonomy

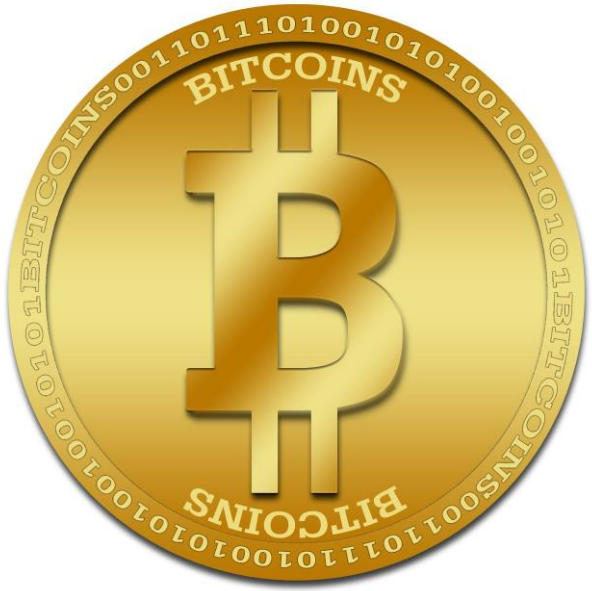


Source: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf

Topic 2: Bitcoin Basics

What Is the Bitcoin?

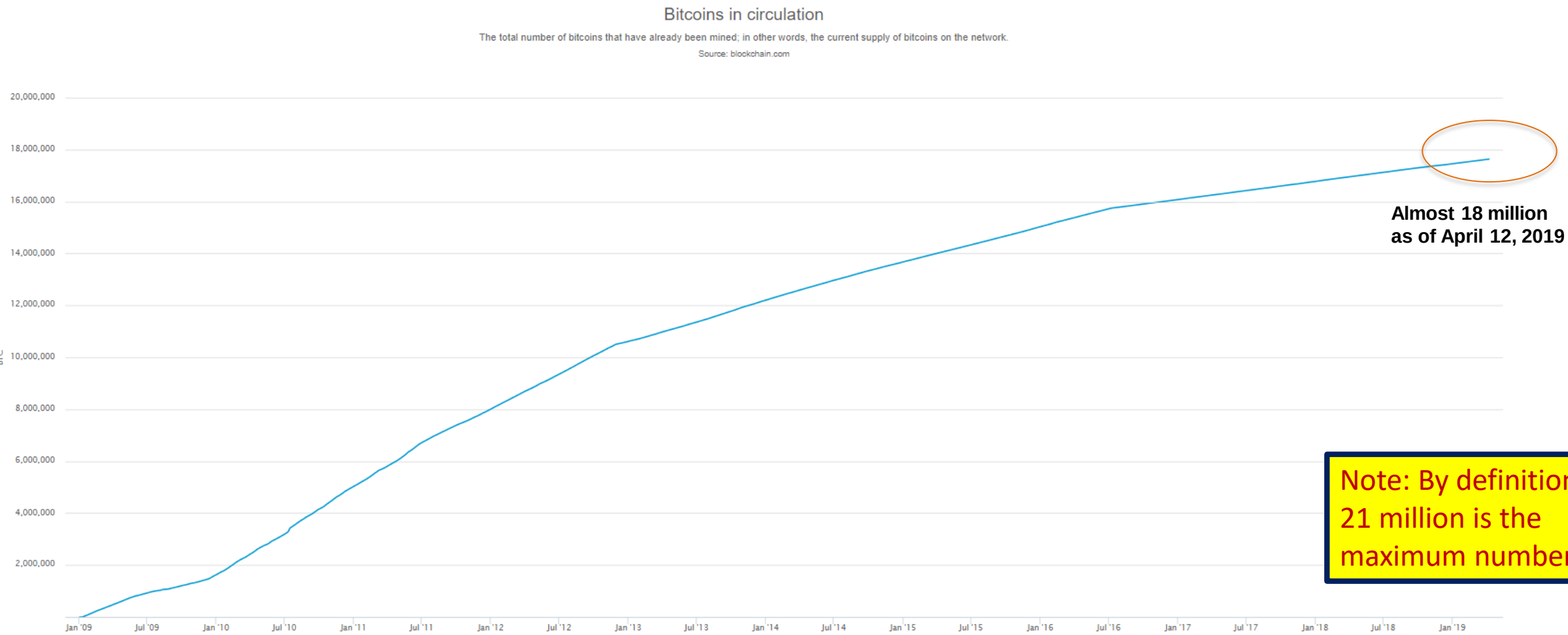
- Digital Currency
- A Decentralized, Peer-to-Peer Payment Network
- Requires the Internet and software to operate
- (Pseudo) Anonymous, untraceable financial transactions
- A standardized “cryptocurrency” that uses a public key and a private key



Bitcoin Characteristics

- **Open Source**
- **Supported by the Bitcoin Foundation**
- **Bitcoin (BTC)**
- http://bitcoin.org/ or http://www.bitcoin.com
- New blocks every **10 min**
- Bitcoin supply **21 million** coins will be available until about 2040
- Difficulty adjustment **1015 blocks, after 6 days**
- Hashing algorithm **SHA256d**
- Initial Reward **50 Bitcoins** per block
- Current reward: **12.5 Bitcoins. In June 2020, it will be halved again to 6.25 Bitcoins**
- Market Cap: $65 Billion (January 2, 2019)
- Over 248,000 Transactions / day
- Launch Date: January 3, 2009

Total Bitcoins in Circulation



Almost 18 million as of April 12, 2019

Note: By definition, 21 million is the maximum number

Source: https://www.blockchain.com/charts/total-bitcoins?timespan=all

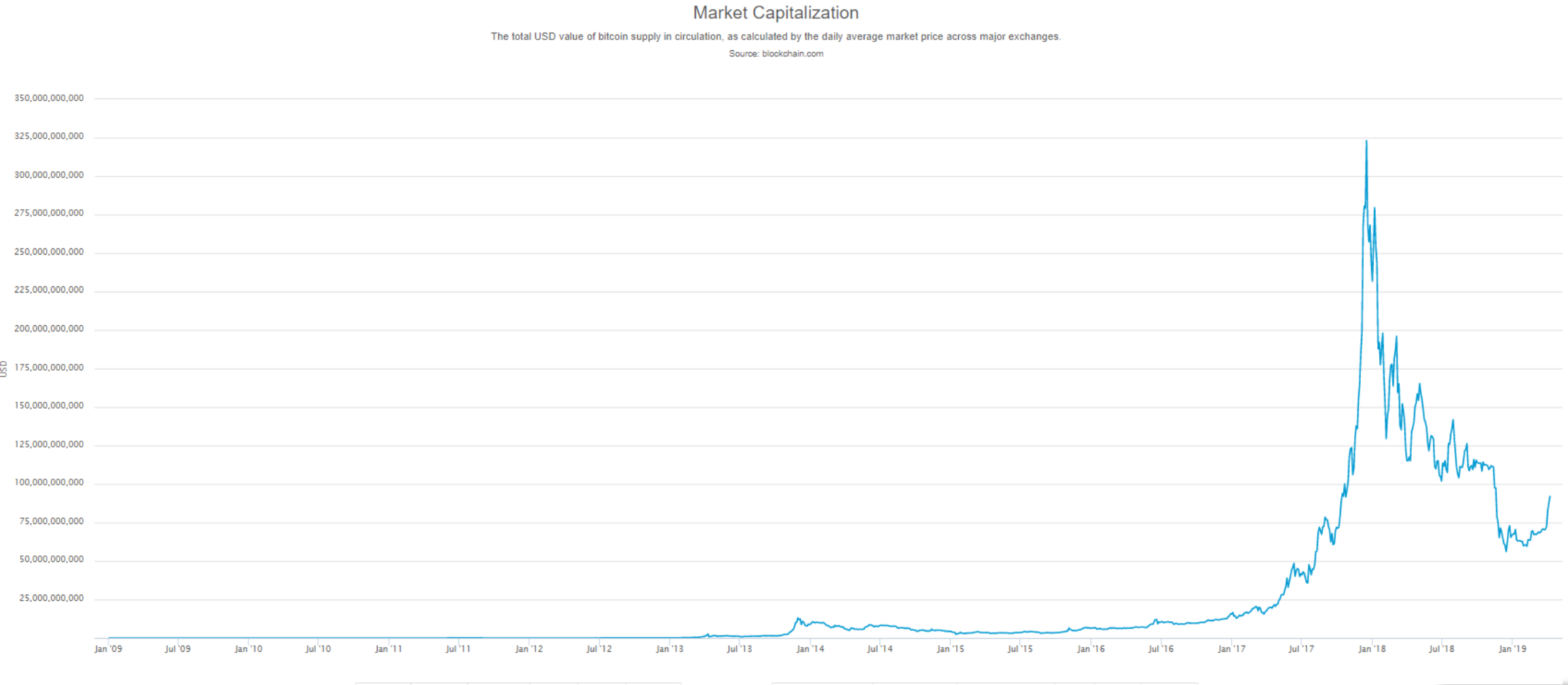
Bitcoin Market Capitalization – 12 Months

Market Capitalization = No. of Bitcoin times the Current Price

Source: https://www.blockchain.com/charts/market-cap

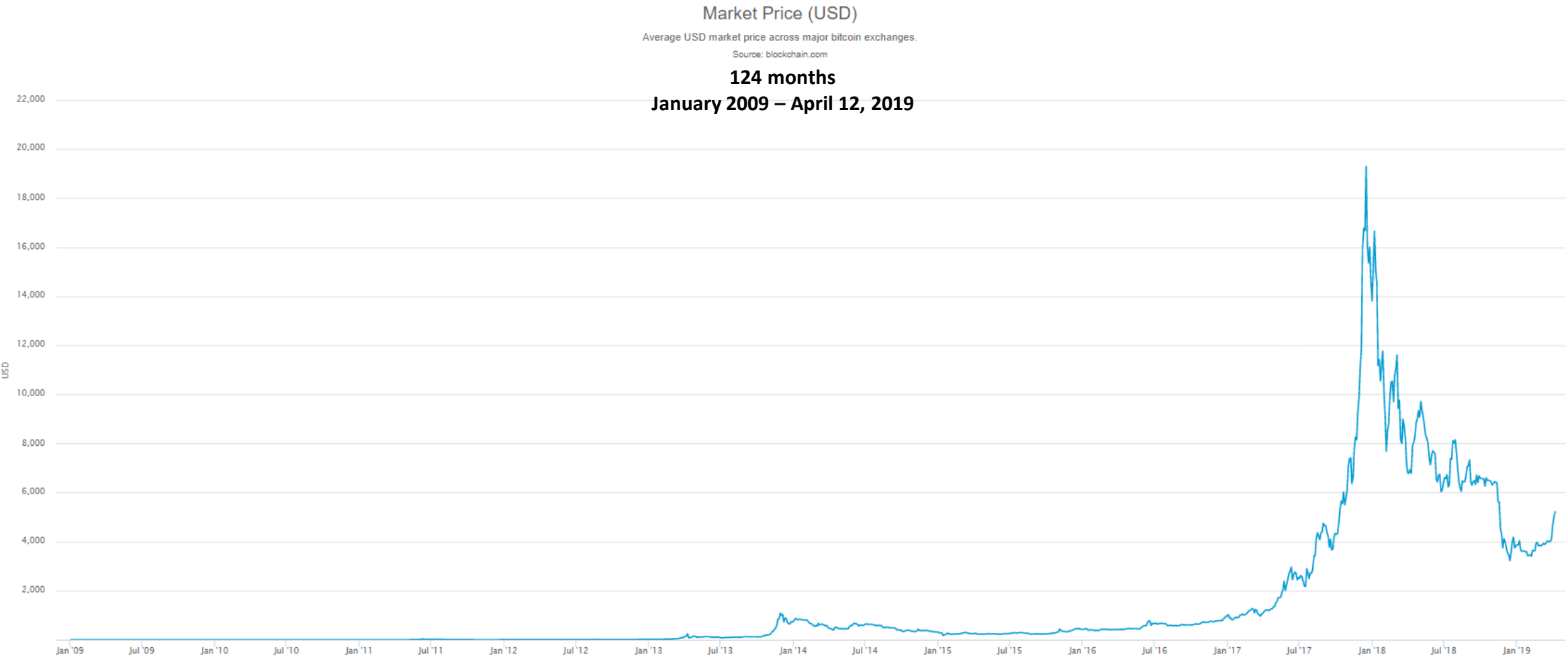
Bitcoin Market Capitalization – All Time

Market Capitalization = No. of Bitcoin times the Current Price



Source: https://www.blockchain.com/charts/market-cap

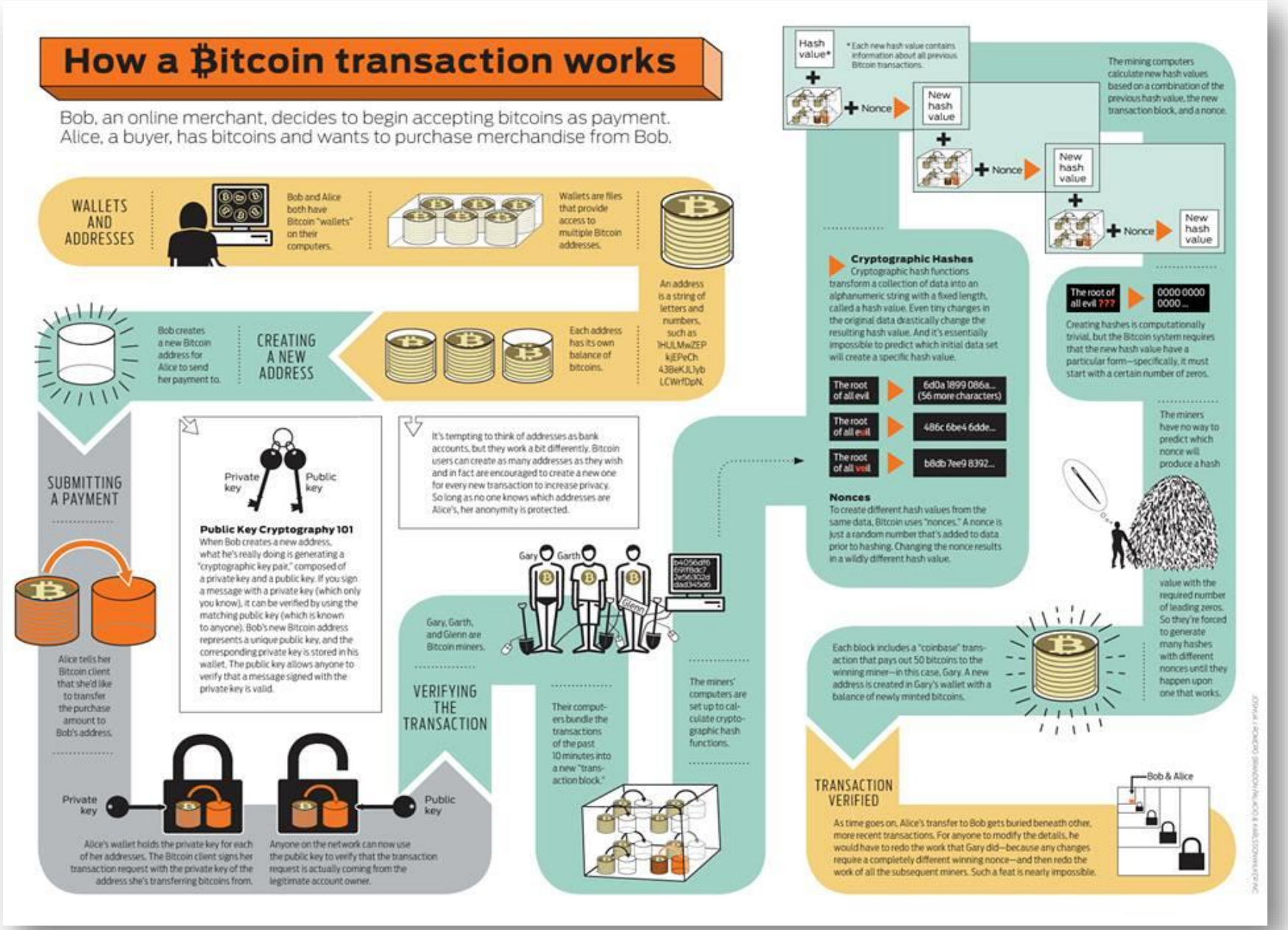
Bitcoin Value in U.S. Dollars

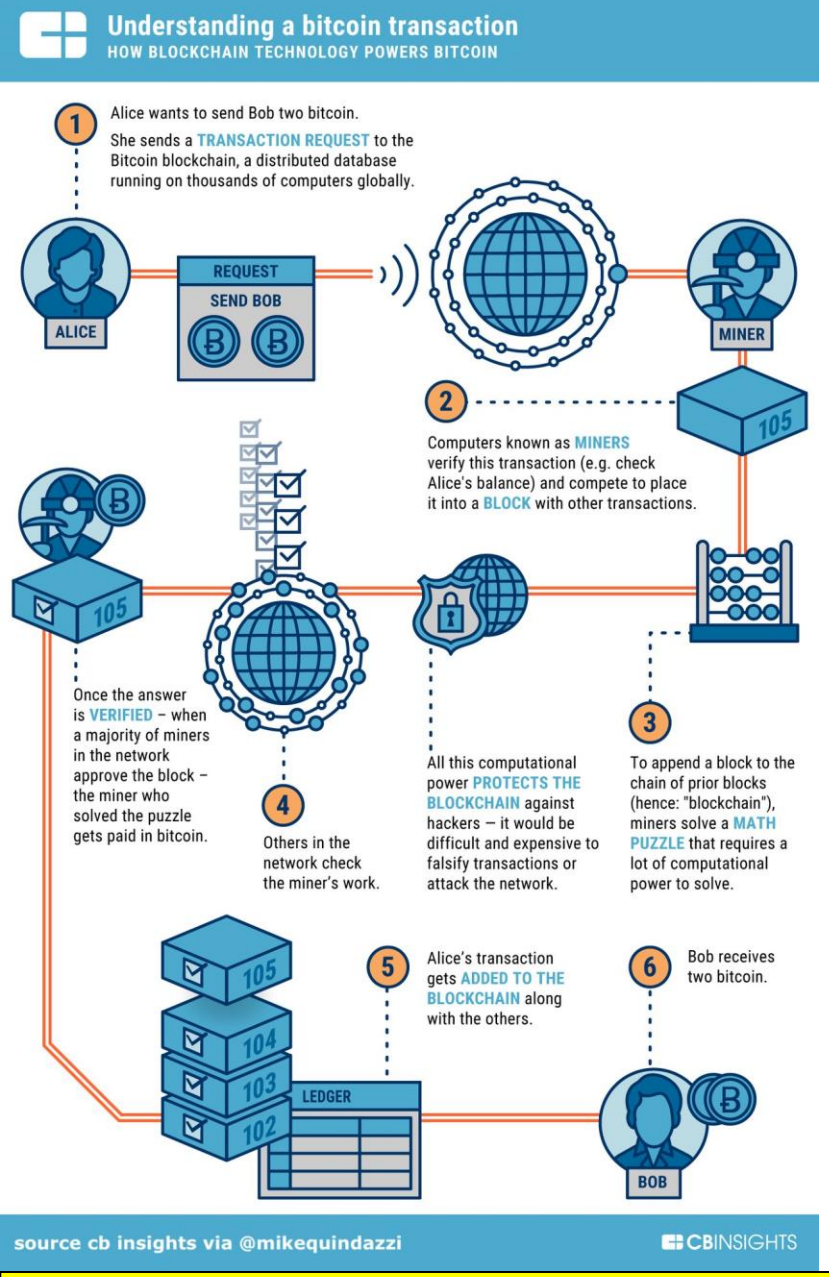


Source: https://www.blockchain.com/charts/market-price?timespan=all

Why Does Bitcoin Have Value/

- Built-in security via its design
- You can buy good and services with it
- Investors speculate in it
- Scarcity
- People (still) believe in it
- Good reputation, mostly
- Technophiles love it
- It’s “cool”





Source: CBInsights.com

Some Bitcoin Terms

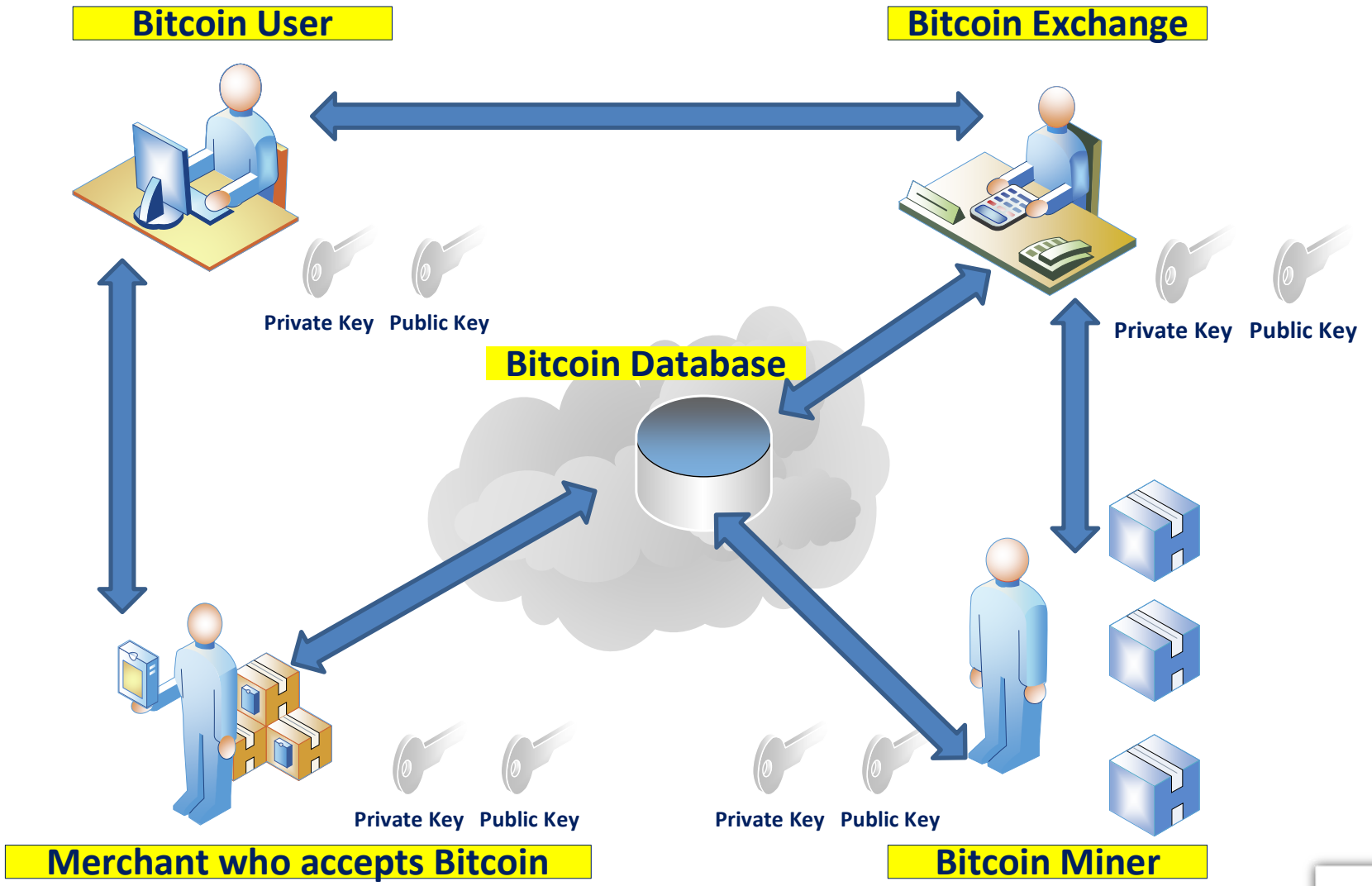
Term	Explanation
AES SHA-256	The 256-bit encryption algorithm that is AES standard used for Bitcoin keys.
Bitcoin Network	The Internet-connected network comprised of the software and data that supports Bitcoin transactioms
Blockchain	The Bitcoin ledger of past transactions.
Difficulty	The measure of how difficult it is to find a new block compared to the easiest it can ever be
Exchange	A place that sells can buys Bitcoins, like a stock exchange.
“Full Node”	A full node is a node that is configured to mine blocks on the blockchain (this applies to Ethereum also)
Hash	It is a standard algorithmic function for the generation and verification of currency
Mining	Bitcoin mining serves 2 purposes, it creates the general ledger of Bitcoin transactions and it provides security.
Private Key	The secret cryptopgraphic key that is used to protect your Bitcoin account
Proof of Work	An economic time-stamped measure to deter service abuses on a network by requiring some work from the service requester, usually meaning processing time by a computer.
Public Key	The public (shared) cryptopgraphic key that is used to protect your Bitcoin account
Transaction	1) Use of the Bitcoin to purchase good or services, or the purchase of sale of a Bitcoin, or fractional part of Bitcoin. 2) Transaction also refers to Blockchain transactions that are stored in Merkel Tree data structures that are hashed and added to each Block in the Blockchain.
Wallet	A service that will safely store your Bitcoin account (public and private keys) for you.

How Does the Bitcoin Network Operate?

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
5. Nodes accept the block only if transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block, using the accepted block as the previous hash.

Source: Bitcoin: A Peer-t-Peer Electronic Cash System by Satoshi Nakamoto https://bitcoin.org/bitcoin.pdf

Bitcoin Actors



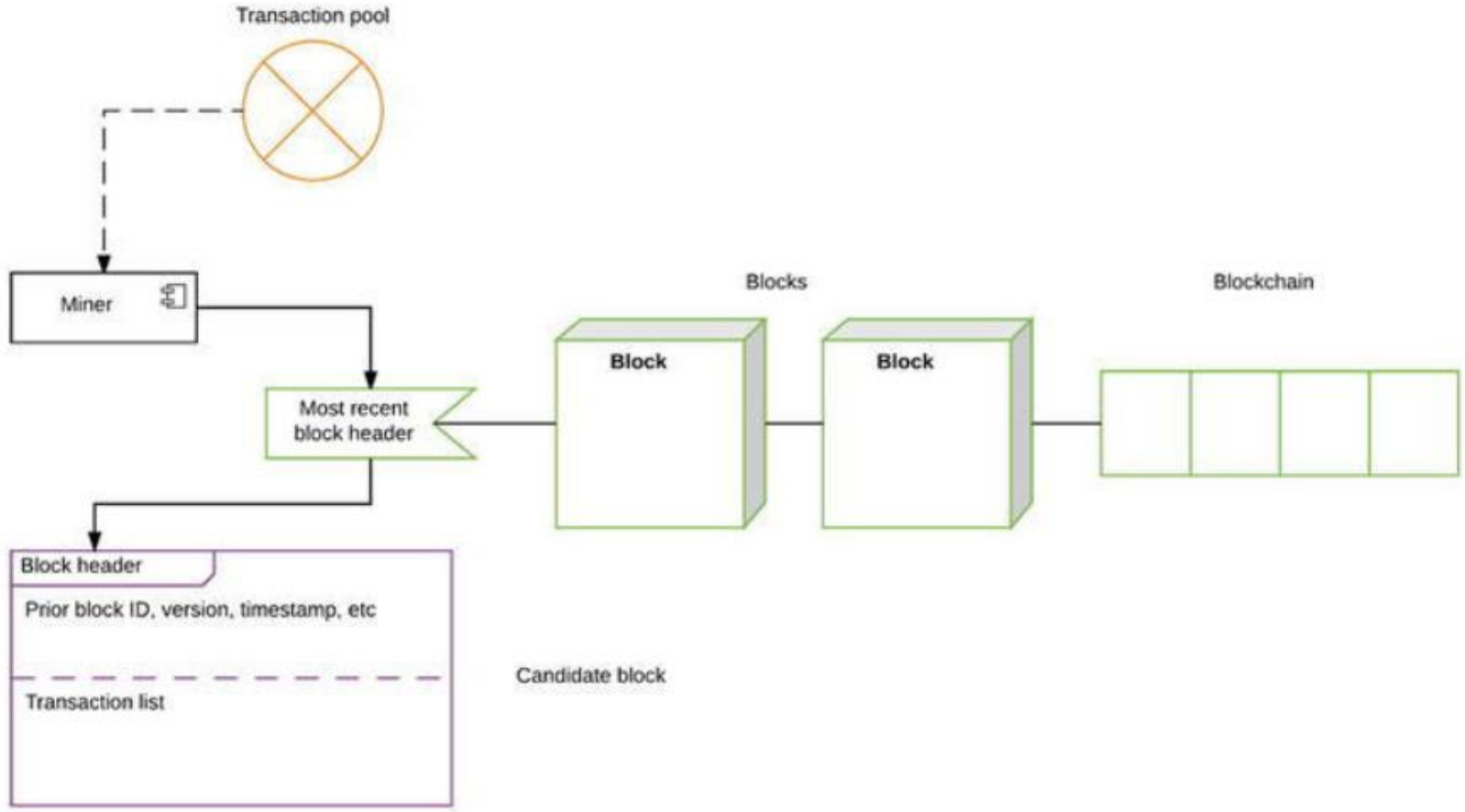
How Does a Bitcoin Trade Work?

- Assume: the Bitcoin user has a legitimate Bitcoin account and knows their balance
- The Bitcoin user finds a business that accepts payments in Bitcoins.
- The Bitcoin user submits their public Bitcoin ID information
- The Bitcoin authorized merchant processes the payment
- The Bitcoin user receives the goods or services

How does Bitcoin Mining Work?

- Mining programs work to perform processing to insert a Bitcoin securely into a valid block chain.
- Processing is very computationally intensive, and uses a lot of CPU time, and a lot of electrical power.
- Rewards:
  - When a block is discovered, the discoverer may award themselves a certain number of bitcoins, which is agreed-upon by everyone in the network. Currently this bounty is 12.5 bitcoins; this value will halve every 210,000 blocks. The next halving is in June 2020, and the new reward will be 6.25 bitcoins.
  - Additionally, the miner is awarded the fees paid by users sending transactions. The fee is an incentive for the miner to include the transaction in their block. In the future, as the number of new Bitcoins miners are allowed to create in each block dwindles, the fees will make up a much more important percentage of mining income.

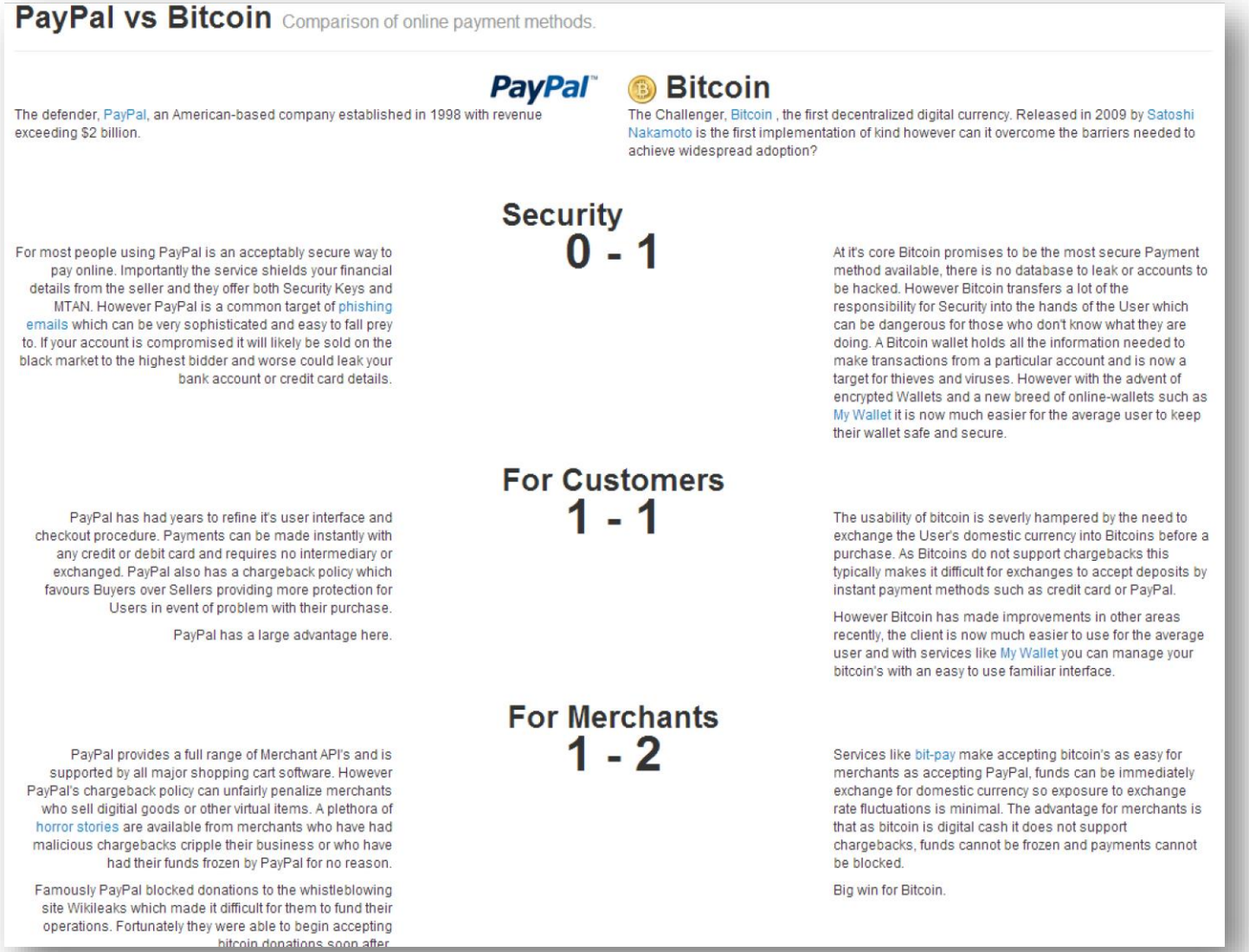
How Does a Bitcoin Mining Work?



A simplified overview of the mining process

Source: **Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You** by Vikram Dhillon, David Metcalf, Max Hooper

Comparing Bitcoin to Paypal



PayPal vs Bitcoin

Comparison of online payment methods.

PayPal

The defender, PayPal, an American-based company established in 1998 with revenue exceeding $2 billion.

Bitcoin

The Challenger, Bitcoin , the first decentralized digital currency. Released in 2009 by Satoshi Nakamoto is the first implementation of kind however can it overcome the barriers needed to achieve widespread adoption?

Security

0 - 1

For most people using PayPal is an acceptably secure way to pay online. Importantly the service shields your financial details from the seller and they offer both Security Keys and MTAN. However PayPal is a common target of phishing emails which can be very sophisticated and easy to fall prey to. If your account is compromised it will likely be sold on the black market to the highest bidder and worse could leak your bank account or credit card details.

At it's core Bitcoin promises to be the most secure Payment method available, there is no database to leak or accounts to be hacked. However Bitcoin transfers a lot of the responsibility for Security into the hands of the User which can be dangerous for those who don't know what they are doing. A Bitcoin wallet holds all the information needed to make transactions from a particular account and is now a target for thieves and viruses. However with the advent of encrypted Wallets and a new breed of online-wallets such as My Wallet it is now much easier for the average user to keep their wallet safe and secure.

For Customers

1 - 1

PayPal has had years to refine it's user interface and checkout procedure. Payments can be made instantly with any credit or debit card and requires no intermediary or exchanged. PayPal also has a chargeback policy which favours Buyers over Sellers providing more protection for Users in event of problem with their purchase.

PayPal has a large advantage here.

The usability of bitcoin is severly hampered by the need to exchange the User's domestic currency into Bitcoins before a purchase. As Bitcoins do not support chargebacks this typically makes it difficult for exchanges to accept deposits by instant payment methods such as credit card or PayPal.

However Bitcoin has made improvements in other areas recently, the client is now much easier to use for the average user and with services like My Wallet you can manage your bitcoin's with an easy to use familiar interface.

For Merchants

1 - 2

PayPal provides a full range of Merchant API's and is supported by all major shopping cart software. However PayPal's chargeback policy can unfairly penalize merchants who sell digitial goods or other virtual items. A plethora of horror stories are available from merchants who have had malicious chargebacks cripple their business or who have had their funds frozen by PayPal for no reason.

Famously PayPal blocked donations to the whistleblowing site Wikileaks which made it difficult for them to fund their operations. Fortunately they were able to begin accepting bitcoin donations soon after

Services like bit-pay make accepting bitcoin's as easy for merchants as accepting PayPal, funds can be immediately exchange for domestic currency so exposure to exchange rate fluctuations is minimal. The advantage for merchants is that as bitcoin is digital cash it does not support chargebacks, funds cannot be frozen and payments cannot be blocked.

Big win for Bitcoin.

Source: https://blockchain.info/wallet/paypal-vs-bitcoin

Comparing Bitcoin to Paypal

For Merchants
1 - 2

PayPal provides a full range of Merchant API's and is supported by all major shopping cart software. However PayPal's chargeback policy can unfairly penalize merchants who sell digitial goods or other virtual items. A plethora of horror stories are available from merchants who have had malicious chargebacks cripple their business or who have had their funds frozen by PayPal for no reason.

Famously PayPal blocked donations to the whistleblowing site Wikileaks which made it difficult for them to fund their operations. Fortunately they were able to begin accepting bitcoin donations soon after.

Services like bit-pay make accepting bitcoin's as easy for merchants as accepting PayPal, funds can be immediately exchange for domestic currency so exposure to exchange rate fluctuations is minimal. The advantage for merchants is that as bitcoin is digital cash it does not support chargebacks, funds cannot be frozen and payments cannot be blocked.

Big win for Bitcoin.

Anonymity
1 - 3

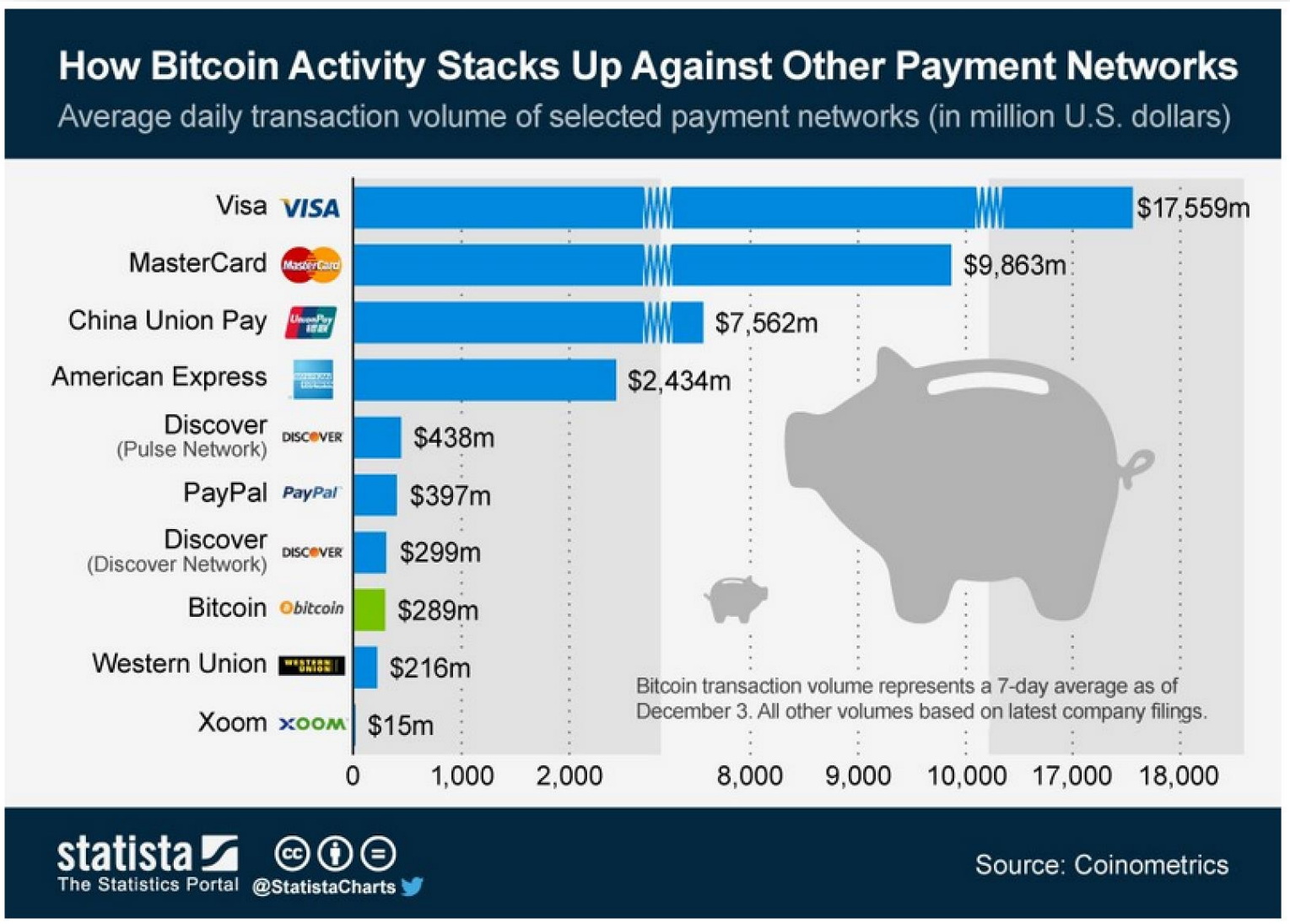
PayPal accounts are tied directly to your bank account or credit card and PayPal is a regulated financial institution in many countries. PayPal payments are not in any way anonymous and it is not recommended you make purchase using PayPal that you would not be confortable with the authorities knowing about.

A history of every bitcoin transaction ever made is available right here on this site. However transactions do not need to be tied to a bank account or individual and they are essentially anonymous if some basic precautions are taken. My Wallet can hold up to 1000 unique bitcoin addresses and it is recommended you change addresses regularly to avoid leaving a trail.

And the winner is. **Bitcoin!**. A new tecnology which is just beginning to come into it's own. Sure there are some hurdles to jump but the ability to truely take control of your own finances is worth some minor inconvienice. If you value liberty, then you should value bitcoin.

Source: https://blockchain.info/wallet/paypal-vs-bitcoin

Comparing Bitcoin to Other Electronic Payment Networks



Source: http://www.businessinsider.com/bitcoin-versus-paypal-comparison-2013-12

Why Is Bitcoin Popular?

- As a cryptocurrency, it has become “the Gold Standard”
- In December 2017, it was valued at about $20,000
- It has made many people, especially young people, millionaires and billionaires
- It’s easily available via the Internet
- International appeal
- It’s “cool”
- It’s supported by many “cool” businesses
- Exciting because it’s in the news
- Anonymous, and uses strong encryption, so it creates a sense of Privacy
- People understand electronic payments because easy to use services like PayPal have been around since 2000

Bitcoin Hype vs. Reality

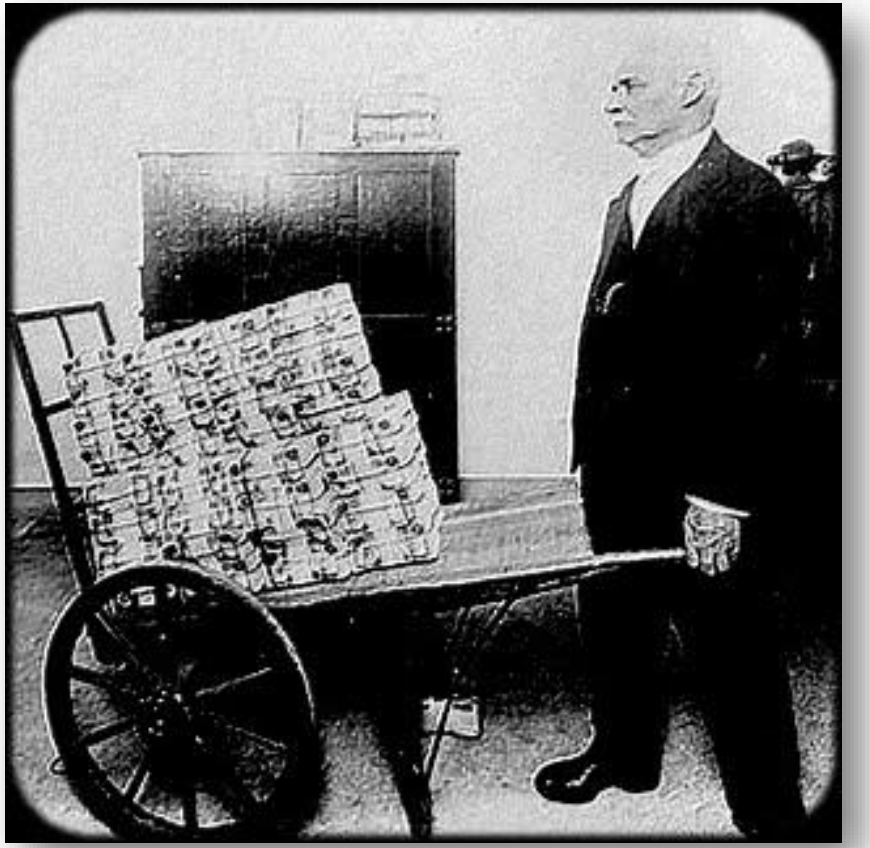
Hype	Reality
Bitcoin is safe	It can be hacked
Bitcoin is anonymous and offers privacy	With entities like the NSA, nothing is or does
Bitcoin is a great investment	No. You can lose money.
Bitcoin mining is lucrative	The IRS is making Retroactive Rulings about Bitcoin as “property”. Talk to your lawyer AND your Accountant.
Bitcoin is simple to use and understand	Do your homework
Bitcoin will become more widely used and accepted	Maybe, but after more than 10 years, it hasn’t happened yet
Bitcoin still has a good name and is widely recognized.	Maybe yes. But events like the Silk Road shutdown, Mt. Gox bankruptcy and Autumn Radtke’s death don’t help Bitcoin’s image

Bitcoin Dangers

- It is still a volatile “investment”
- Vulnerability to Hackers
- Anonymous cryptocurrency transactions can and will arouse suspicion
- No central authority to regulate it
- Not insured
- Some experts have developed an extensive case AGAINST investing in Bitcoin
- The PBOC banned Bitcoin expenditures, but not mining inside China – 2014
- The IRS is regulating it retroactively – Virtual Currency Guidance – March 25, 2014
- Unsecure Wallets are a Huge Knoown Vulnerability
- Opinion: Buying Bitcoin when it is above $15,000

Bitcoin and the Future of the Global Economy

- The increasing visibility and acceptance of Bitcoin have given it positive international recognition
- Increasing concerns about the stability of U.S. Dollar and other fiat currencies (inflation, hyperinflation, debt, etc.), as well as geopolitical uncertainties have caused speculation in unusual investments like Bitcoin
- The ever-increasing population of younger investors that understand accumulation and management of wealth better, as well as Bitcoin and Cryptocurrency portend a bright future for Bitcoin.
- There are over 83 million Millennials.



Hyperinflation in Germany in 1923

Bitcoin Conclusion

Bitcoin:

- A technical marvel made possible by software, hardware, strong cryptography, and the Internet
- Has made significant progress in only 124 months
- Has significant strengths and weaknesses
- Has great potential because of popular support of talented nerds
- Has attracted the interest of those who would like to control it (U.S. Government, especially the IRS)
- Should be watched, studied, and understood carefully before making any big investments in Bitcoin accounts, mining, accepting transactions, etc.

Topic 3: The Tokenized Economy and Cryptocurrency Concepts

The Tokenized Economy and Cryptocurrency Concepts

- A token is a privately issued cryptocurrency.
- In the business realm, we can define a token as: "A unit of value that an organization creates to self-govern its business model, and empower its users to interact with its products, while facilitating the distribution and sharing of rewards and benefits to all of its stakeholders."
- The Achilles heel of token-based models will be how they are concocted to interact with the business model that underlies them. However, much of the attention has been on designing ICO's to optimize for cryptoeconomics, a term that has come to describe the mechanics and specifics of token distribution, according to a given sale and ownership structure.
- **Good News:** Tokenization allows tangible things like real estate, art, etc. to be catalogued and traded using Blockchain and Cryptocurrency technologies.
- **Bad News:** Between 2017 and 2018, ICOs got a very bad name because so many were issued and ultimately mismanaged and failed. Many people in the market decided the hype and risks were not worth the potential rewards.

Source: https://medium.com/@wmougayar/tokenomics-a-business-guide-to-token-usage-utility-and-value-b19242053416

ERC Tokens

ERC20 token is an interface which defines various functions dictating the requirements of the token. It does not, however, provide implementation details and has been left to the implementer to decide. ERC is basically an abbreviation of **Ethereum Request for Comments** which is equivalent to Bitcoin's BIPs for suggesting improvements in Ethereum blockchain.

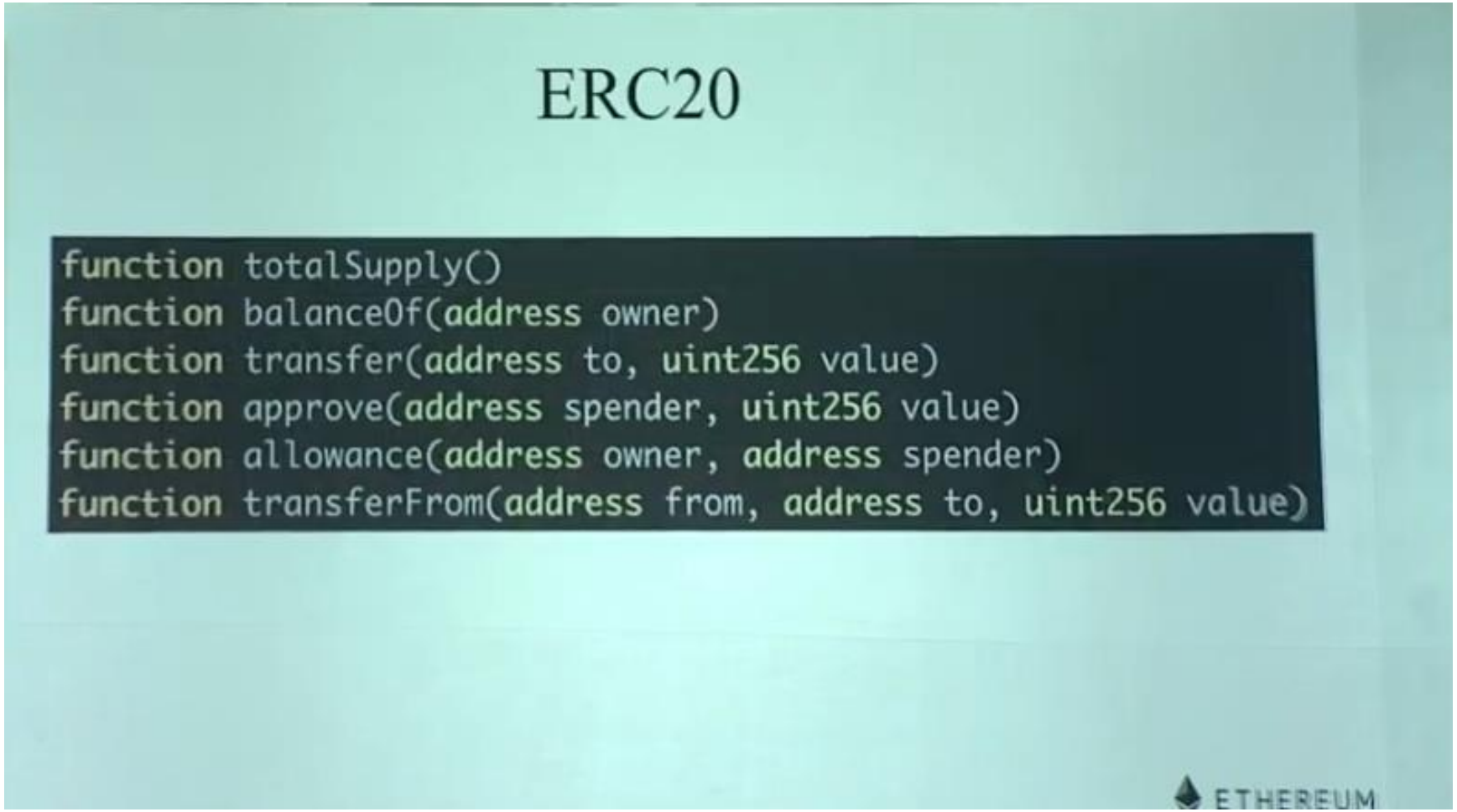
> This is defined under EIP 20, which you can read more about here `https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20-token-standard.md`.

Ethereum is becoming a platform for choice for ICOs due to its ability to create new tokens and with ERC20 standard, it has become even more accessible.

ECR20 token standard defines various functions which describe various properties, rules, and attributes of the new token. These include total supply of the coins, total balance of holders, transfer function, approval and allowance functions.

Source: Mastering Blockchain by Imran Bashir (Published by Packt.)

ERC 20



Source: Ethereum Foundation - Hudson Jameson - https://www.youtube.com/watch?v=KkN1O8TChbM

ERC Tokens

- ERC-721 is the standard for Ethereum tokens that are not related to cryptocurrency.
- They are non-fungible
- http://erc721.org/
- https://medium.com/@brenn.a.hill/noobs-guide-to-understanding-erc-20-vs-erc-721-tokens-d7f5657a4ee7

Source: Mastering Blockchain by Imran Bashir (Published by Packt.)

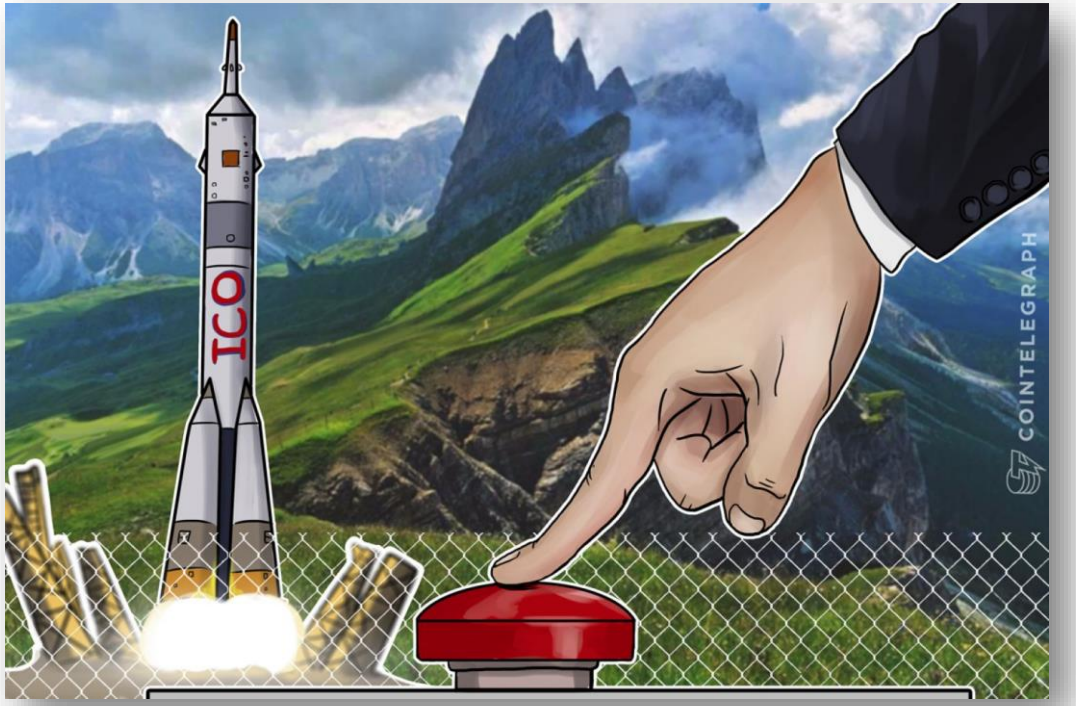
12 Steps to Do an ICO

1. Review other ICOs to see what you need to work on
2. Creating white paper
3. Building the Team
4. Resolving your legal questions
5. Preparing your website
6. Security of your ICO platform
7. Other security considerations
8. Hiring your smart contract developer and auditing their work
9. Ensure that your crowdfunding campaign follows common rules
10. Announce your ICO
11. Your social media and communication channels
12. PR and Marketing

Source: Bogdan Fiedur. April 5, 2018 https://hackernoon.com/how-to-start-your-own-ico-when-you-are-new-to-crypto-12-proven-steps-edc09f25ad66

8 Steps to Launch an ICO

1. Come up with an idea, think it through and make sure that your project actually needs an ICO
2. Know your competition
3. Research the legal side of things and find out if ICOs are actually legal in your country ICOs are currently the most regulated aspect of cryptocurrencies there is.
4. Create an ICO token
5. Write a white paper
6. Launch a website
7. PR and marketing
8. Launch an ICO



Source: CoinTelegraph. https://cointelegraph.com/ico-101/how-to-launch-an-ico-a-detailed-guide#1-come-up-with-an-idea-think-it-through-and-make-sure-that-your-project-actually-needs-an-icopril 5, 2018

16 Steps to Do an ICO

1. Formulate the idea
2. Assemble the team
3. Examine the competitors
4. Register the company
5. Describe the product based on the idea (Whitepaper)
6. Launch the site and email campaigns
7. Describe the conditions for investors.
8. Create social channels
9. Develop and publish a bounty-campaign
10. Place the project in the ICO trackers.
11. Place the materials in thematic media
12. Launch advertising
13. Develop an investor's personal cabinet
14. Make translations into other languages
15. Issue Tokens
16, Start ICO

Source: Bogdan Fiedur. April 5, 2018 https://hackernoon.com/how-to-start-your-own-ico-when-you-are-new-to-crypto-12-proven-steps-edc09f25ad66

Tokenized Economy: 20 Questions for an ICO to Answer

Questions 1 - 10

1. Is the token tied to a product usage, i.e. does it give the user exclusive access to it, or provide interaction rights to the product?
2. Does the token grant a governance action, like voting on a consensus related or other decision-making factor?
3. Does the token enable the user to contribute to a value-adding action for the network or market that is being built?
4. Does the token grant an ownership of sorts, whether it is real or a proxy to a value?
5. Does the token result in a monetizable reward based on an action by the user (active work)?
6. Does the token grant the user a value based on sharing or disclosing some data about them (passive work)?
7. Is buying something part of the business model?
8. Is selling something part of the business model?
9. Can users create a new product or service?
10. Is the token required to run a smart contract or to fund an oracle? (an oracle is a source of information or data that other than a smart contract can use)

Source: https://medium.com/@wmougayar/tokenomics-a-business-guide-to-token-usage-utility-and-value-b19242053416

Tokenized Economy: 20 Questions for an ICO to Answer

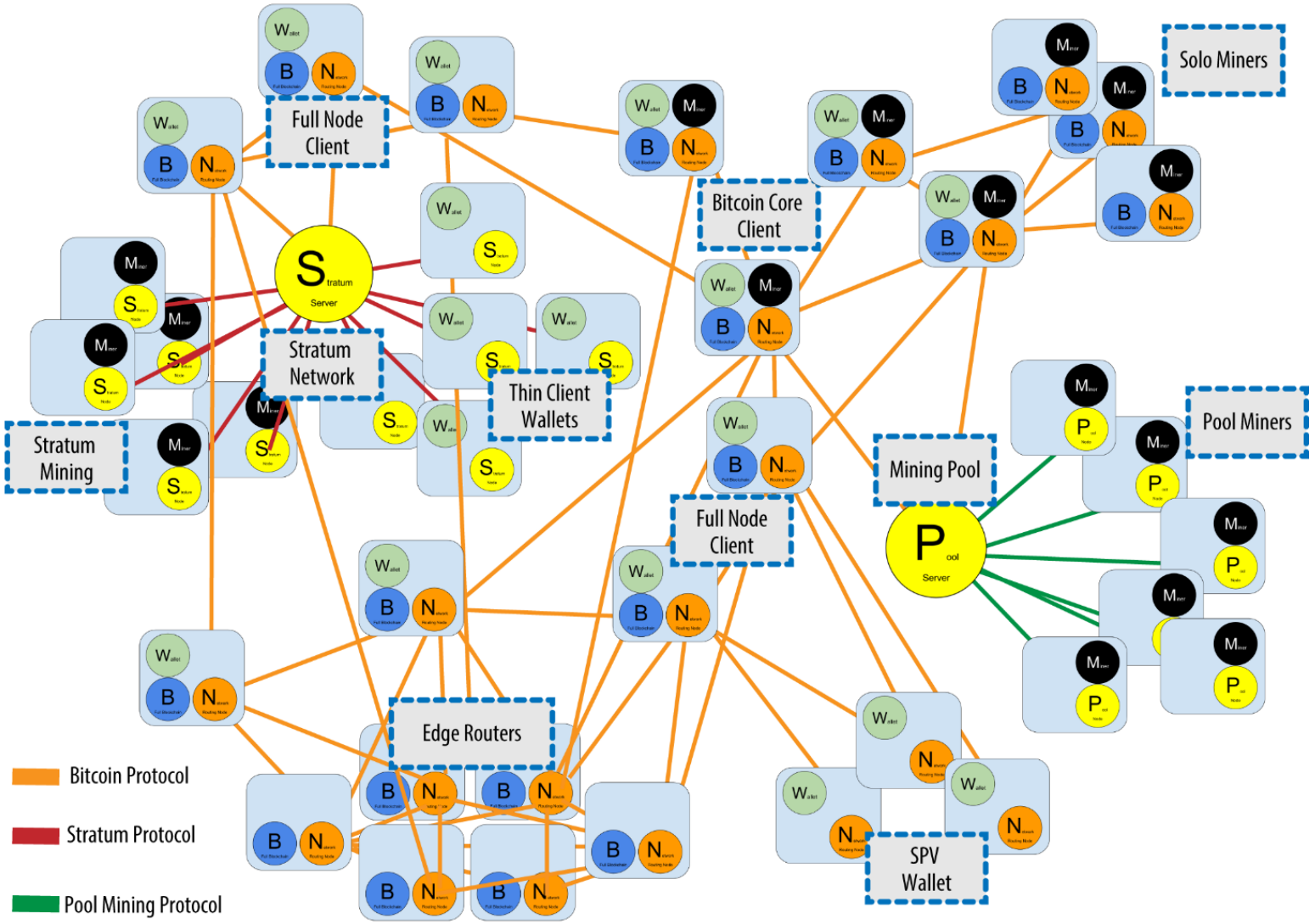
Questions 11 - 20

11. Is the token required as a security deposit to secure some aspect of the blockchain's operation?
12. Is the token (or a derivative of it, like a stable coin or gas unit) used to pay for some usage?
13. Is the token required to join a network or other related entity?
14. Does the token enable a real connection between users?
15. Is the token given away or offered at a discount, as an incentive to encourage product trial or usage?
16. Is the token your principal payment unit, essentially functioning as an internal currency?
17. Is the token (or derivative of it) the principal accounting unit for all internal transactions?
18. Does your blockchain autonomously distribute profits to token holders?
19. Does your blockchain autonomously distribute other benefits to token holders?
20. Is there a related benefit to your users, resulting from built-in currency inflation?

Source: https://medium.com/@wmougayar/tokenomics-a-business-guide-to-token-usage-utility-and-value-b19242053416

Topic 4: Blockchain Technology

A Logical Diagram of a Blockchain Network



What Is Blockchain?

- Distributed Ledger
- Decentralized
- Popularized by Satoshi Nakamoto (Bitcoin inventor)
- Uses Public-Key Cryptography and Hashing
- Append-only Transactions
- The Open Source Code already exists in Github (Bitcoin and Ethereum)
- Immutable (cannot delete blocks or change data in blocks)
- Driven by consensus protocol(s)
  - Proof of Work
  - Proof of Stake
  - Etc.
- The world's largest Blockchain Database is the Bitcoin Blockchain Database, with 180 GB (it doesn't scale very well)

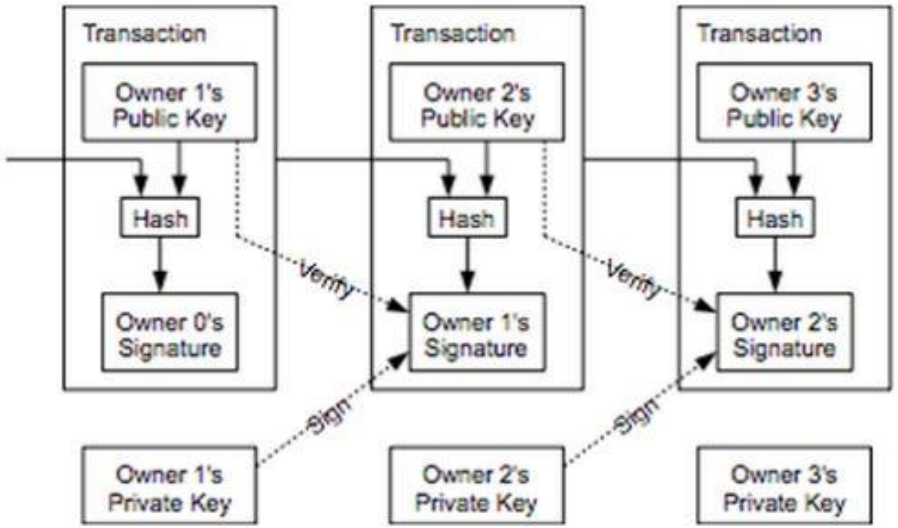


Image: Satoshi Nakamoto

What Is Blockchain?

- **From Blockchain Consensus Protocol Guide:**
- A blockchain is a decentralized peer-to-peer system with no central authority figure.
- While this creates a system that is devoid of corruption from a single source, it still create a major problems:
    - How are any decisions made?
    - How does anything get done?
    - Think of a normal centralized organization.
- All the decisions are taken by the leader or a board of decision makers. This isn't possible in a blockchain because a blockchain has no "leader". For the blockchain to make decisions, they need to come to a consensus using "consensus mechanisms".

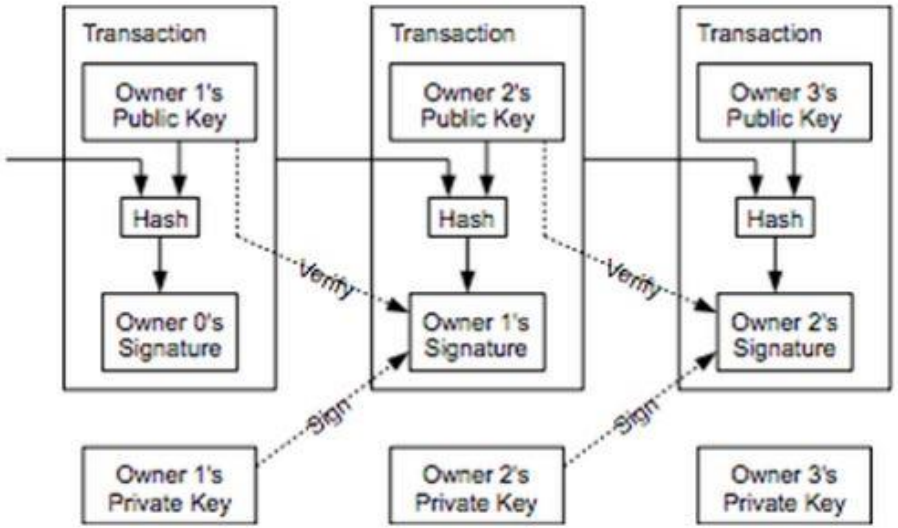


Image: Satoshi Nakamoto

The Term “Blockchain”

- Name for a data structure
- Name for an algorithm
- Name for a suite of Technologies
- An umbrella term for purely distributed peer-to-peer systems with a common application area
- A peer-to-peer-based operating system with its own unique rule set that utilizes hashing to provide unique data

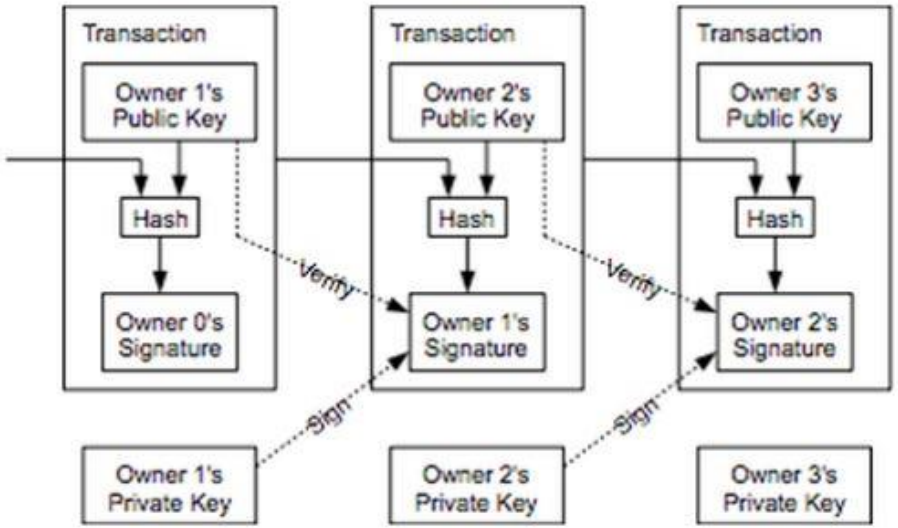


Image: Satoshi Nakamoto

Blockchain – Simplified View

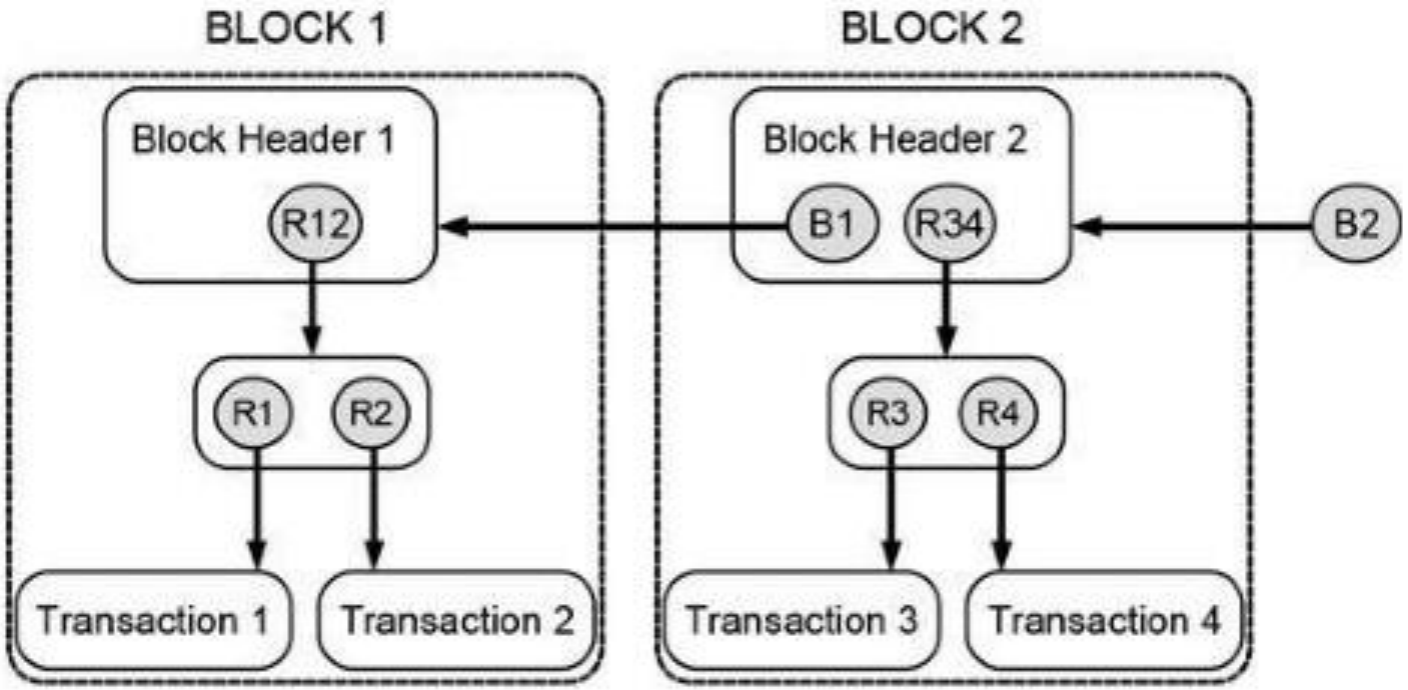


Figure 14-5. A simplified blockchain-data-structure containing four transactions

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

The Term “Blockchain”

The blockchain is a purely distributed peer-to-peer data store with the following properties:

- Immutable
- Append-only
- Ordered
- Time-stamped
- Open and transparent
- Secure (identification, authentication, and authorization)
- Eventually consistent

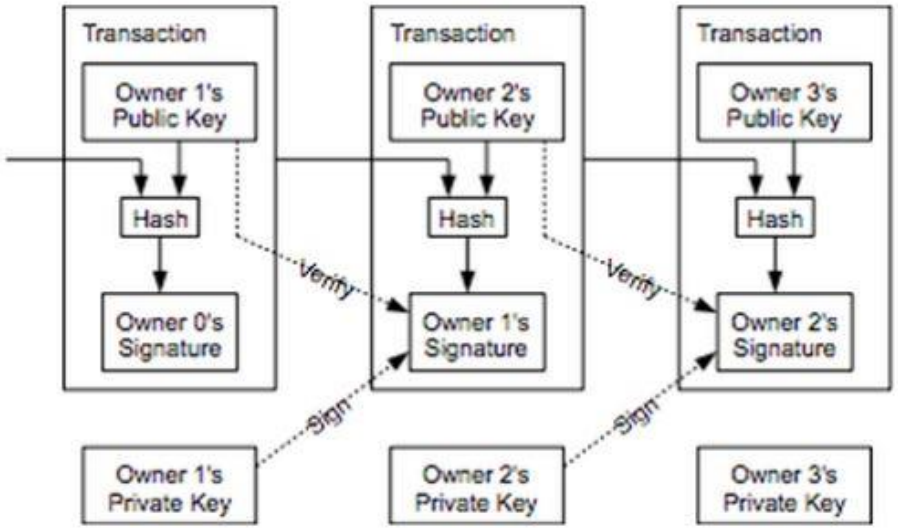


Image: Satoshi Nakamoto

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

Properties on the Blockchain's Non-Functional Aspects

When interacting with the blockchain, you will notice how it fulfills its duties. The quality at which the blockchain serves its purpose is described by its nonfunctional aspects:

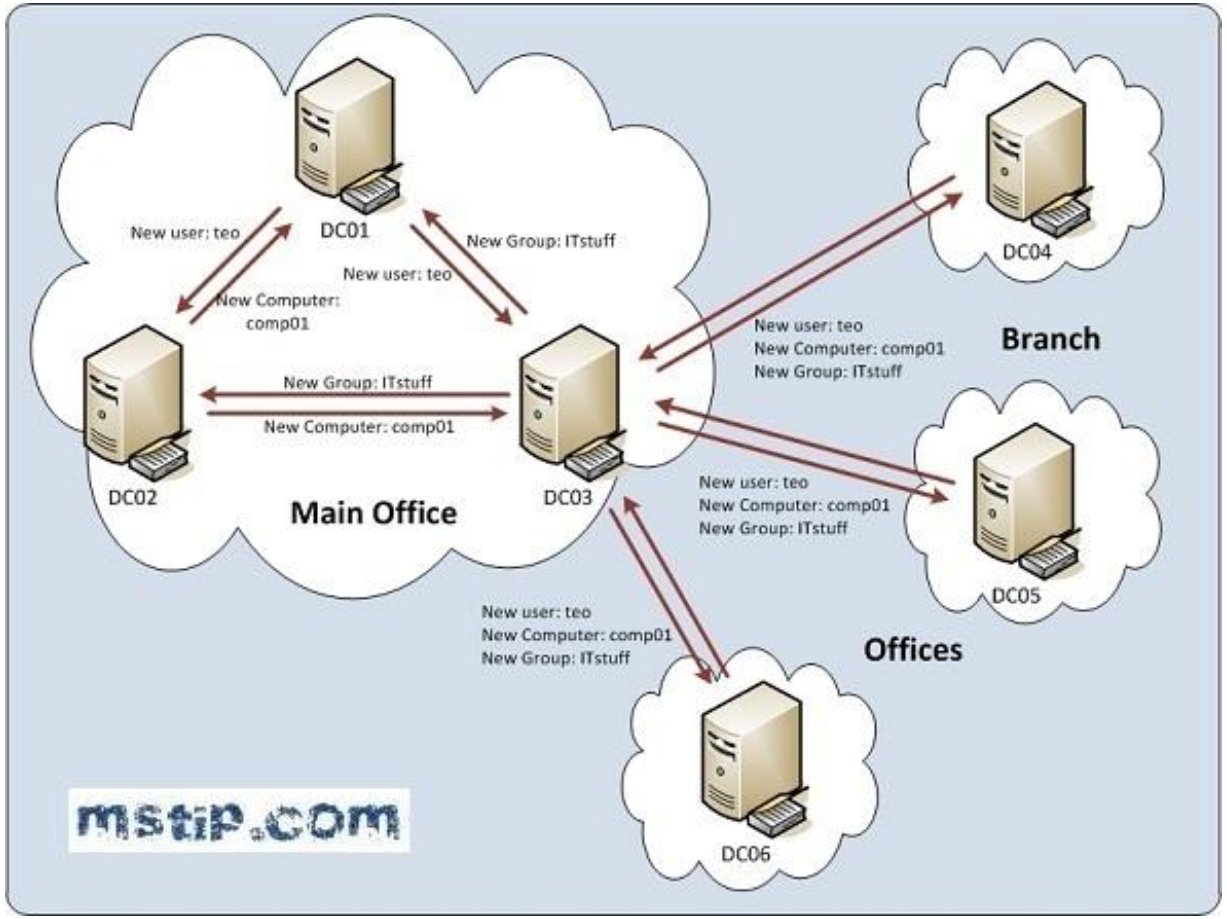
- Highly available
- Censorship proof
- Reliable
- Open
- Pseudoanonymous
- Secure
- Resilient
- Eventually consistent

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

Analogy for Blockchain Updates

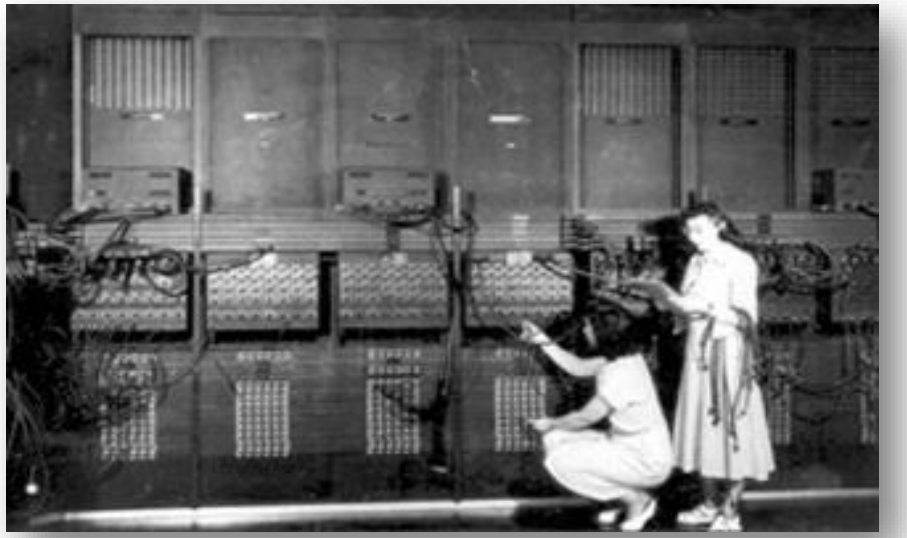
Like Windows Active Directory replicating data on Domain Controllers via The Knowledge Consistency Checker algorithm, Blockchain miner nodes and client are updated with the latest Block each time a consensus is agreed upon.

Except all nodes on a Blockchain (full-nodes and clients) are constantly updated with current Block information.



Technologies and Events that Led to the Creation of Bitcoin and Blockchain

- Cryptography
- Transistors
- Digital Computers
- Databases
- Silicon Chips
- Programming
- Applied Cryptography
- Computer Networks
- Transaction Processing
- TCP/ IP and The Internet
- The World Wide Web
- Evolution of Security and Privacy Thought
- Digital signatures
- Time-stamped documents
- Smart Contracts
- Byzantine Fault Tolerance
- The Great 2008 Economic Recession



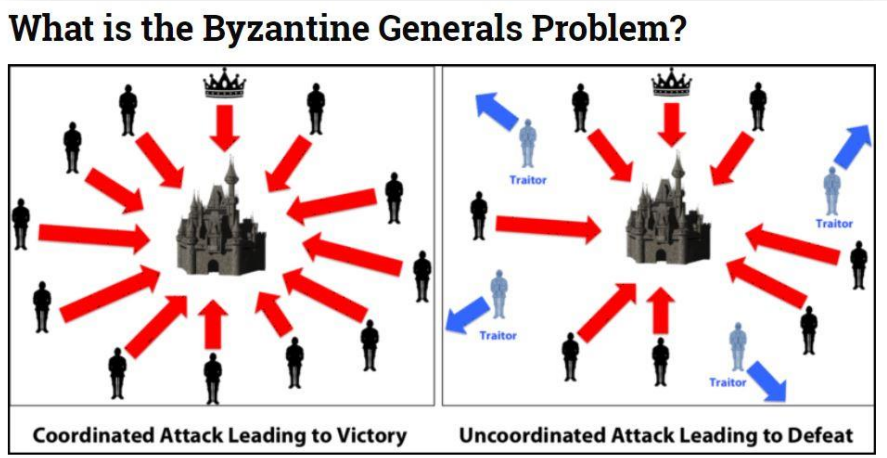
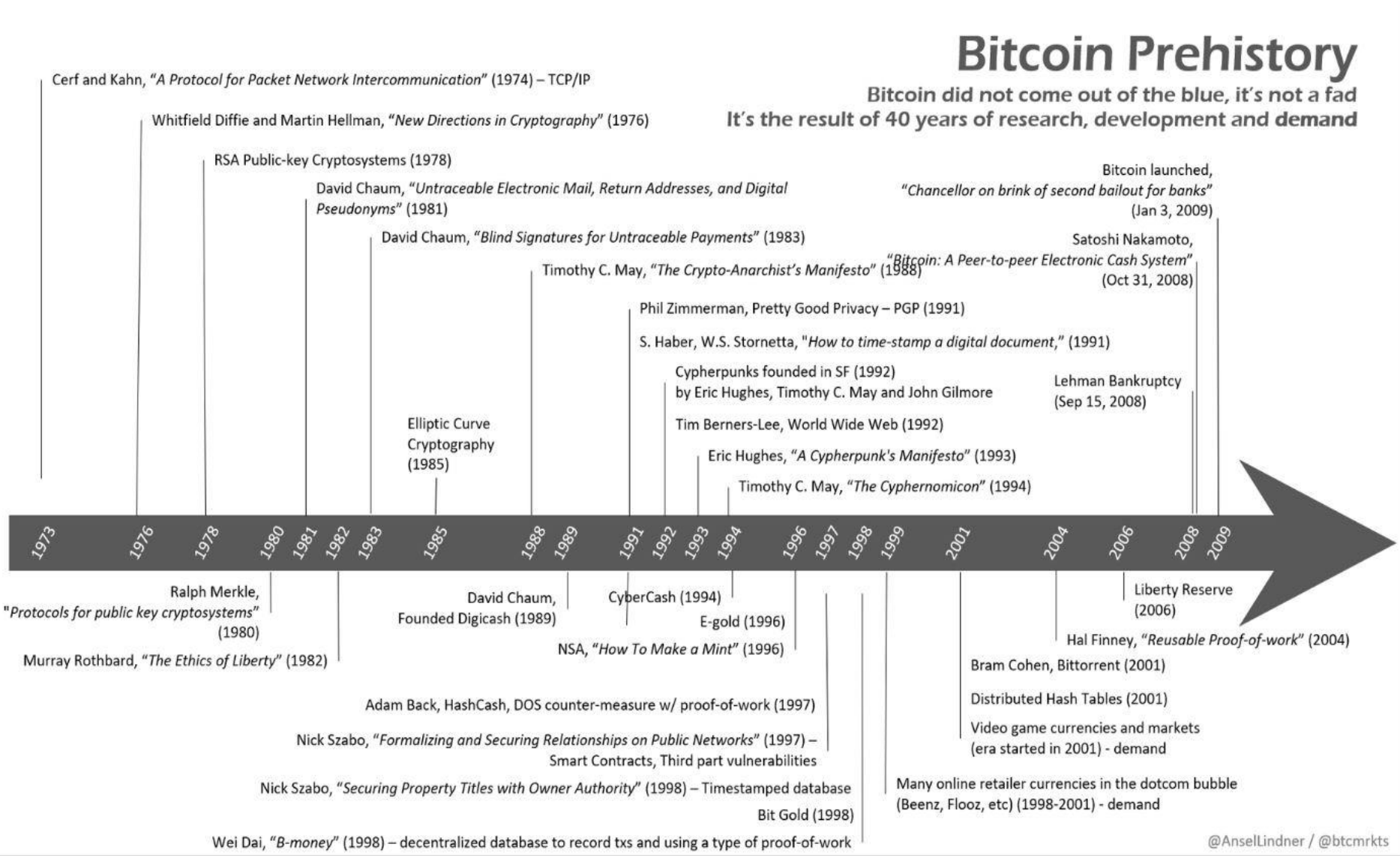


Image Courtesy: Medium

Technologies and Events that Led to the Creation of Bitcoin and Blockchain



Blockchain Technologies

Technology	Source
The Internet (TCP/IP)	Built into every modern OS
Cryptography	Cryptography software
Bitcoin software	Github
Ethereum Software (geth)	Github
Blockchain Database	JSON (default), Bigchain, NEM, Factom, etc.

Authentication in the Blockchain

Authentication in the Blockchain

- Identifying accounts: User accounts are public cryptographic keys.
- Authorizing transactions: The owner of the account who hands off ownership creates a piece of cypher text with the corresponding private key. This piece of cypher text can be verified by using the corresponding public key, which happens to be the number of the account that hands off ownership.

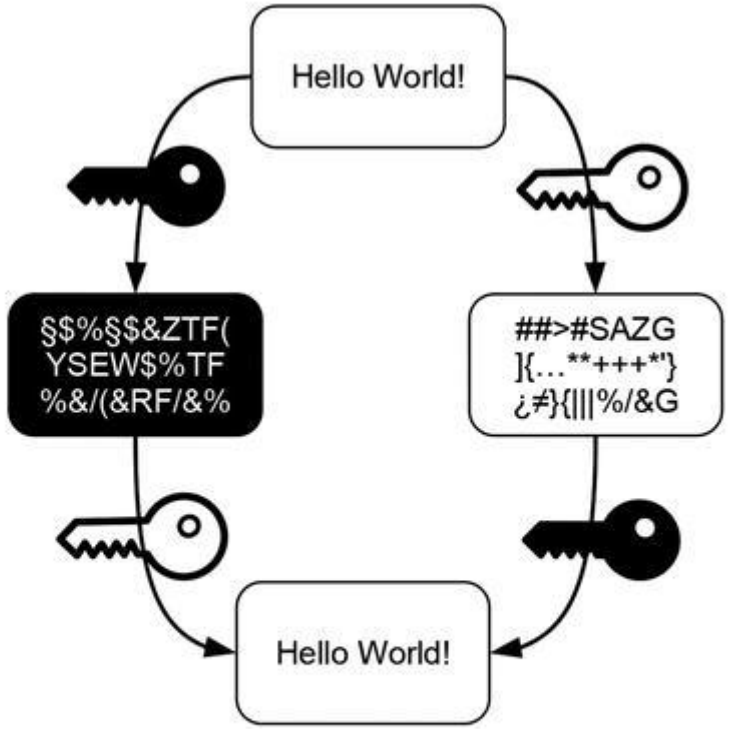


Figure 12-3. Schematic illustration of asymmetric cryptography

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

Types of Blockchains

Types of Blockchains

Bitcoin vs. Ethereum vs. Hyperledger (Linux and IBM) and now many others

Public vs. Private

Permissioned (private) vs. Permissionless

Bitcoin vs. Ethereum

VS	Bitcoin	Ethereum
Founder	Satoshi Nakamoto	Vitalik Buterin
Release Date	9 Jan 2008	30 July 2015
Release Method	Genesis Block Mined	Presale
Blockchain	Proof of work	Proof of work (Planning for POS)
Useage	Digital Currency	Smart Contracts Digital Currency
Cryptocurrency Used	Bitcoin(Satoshi)	Ether
Algorithm	SHA-256	Ethash
Blocks Time	10 Mintues	12-14 Seconds
Mining	ASIC miners	GPUs
Scalable	Not now	Yes

Comparing Ethereum, Hyperledger, and Cordia

Comparison of Ethereum, Hyperledger Fabric and Corda

Characteristic	Ethereum	Hyperledger Fabric	R3 Corda
Description of platform	– Generic blockchain platform	– Modular blockchain platform	– Specialized distrib-uted ledger platform for financial industry
Governance	– Ethereum developers	– Linux Foundation	– R3
Mode of operation	– Permissionless, public or private[4]	– Permissioned, private	– Permissioned, private
Consensus	– Mining based on proof-of-work (PoW) – Ledger level	– Broad understand-ing of consensus that allows multiple approaches – Transaction level	– Specific understand-ing of consensus (i.e., notary nodes) – Transaction level
Smart contracts	– Smart contract code (e.g., Solidity)	– Smart contract code (e.g., Go, Java)	– Smart contract code (e.g., Kotlin, Java) – Smart legal contract (legal prose)
Currency	– Ether – Tokens via smart contract	– None – Currency and tokens via chaincode	– None

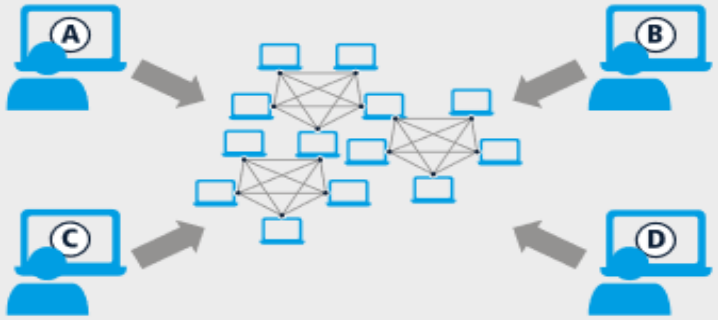
Comparing Bitcoin, Ethereum, & Hyperledger

Blockchain characteristics comparison

Characteristics	Bitcoin	Ethereum	Hyperledger
Permission restrictions	Permissionless	Permissionless	Permissioned
Restricted public access to data	Public	Public or private	Private
Consensus	Proof-of-Work	Proof-of-Work	PBFT
Scalability	High node-scalability, Low performance-scalability	High node-scalability, Low performance-scalability	Low node-scalability, High performance-scalability
Centralized regulation (governance*)	Low, decentralized decision making by community/miners	Medium, core developer group, but EIP process	Low, open-governance model based on Linux model
Anonymity	Pseudonymity, no encryption of transaction data	Pseudonymity, no encryption of transaction data	Pseudonymity, encryption of transaction data
Native currency	Yes, bitcoin, high value	Yes, ether	No
Scripting	Limited possibility, stack-based scripting	High possibility, Turing-complete virtual machine, high-level language support (Solidity)	High possibility, Turing-complete scripting of chaincode, high-level Go-language

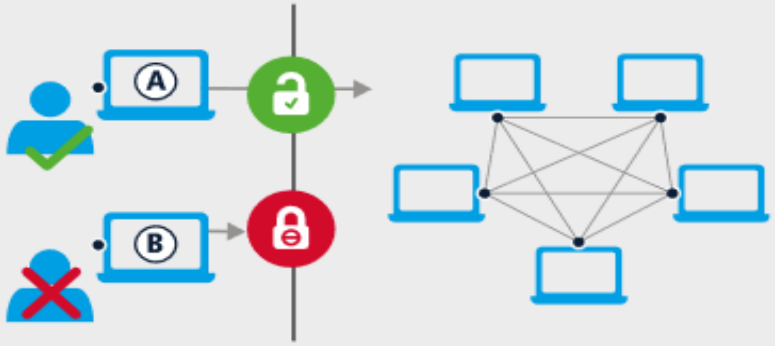
Public vs. Private

PUBLIC VS. PRIVATE BLOCKCHAINS



PUBLIC, PERMISSIONLESS BLOCKCHAINS

- Anyone can join the network and submit transactions
- Anyone can contribute computing power to the network and broadcast network data
- All transactions are broadcast publicly



PRIVATE, PERMISSIONED BLOCKCHAINS

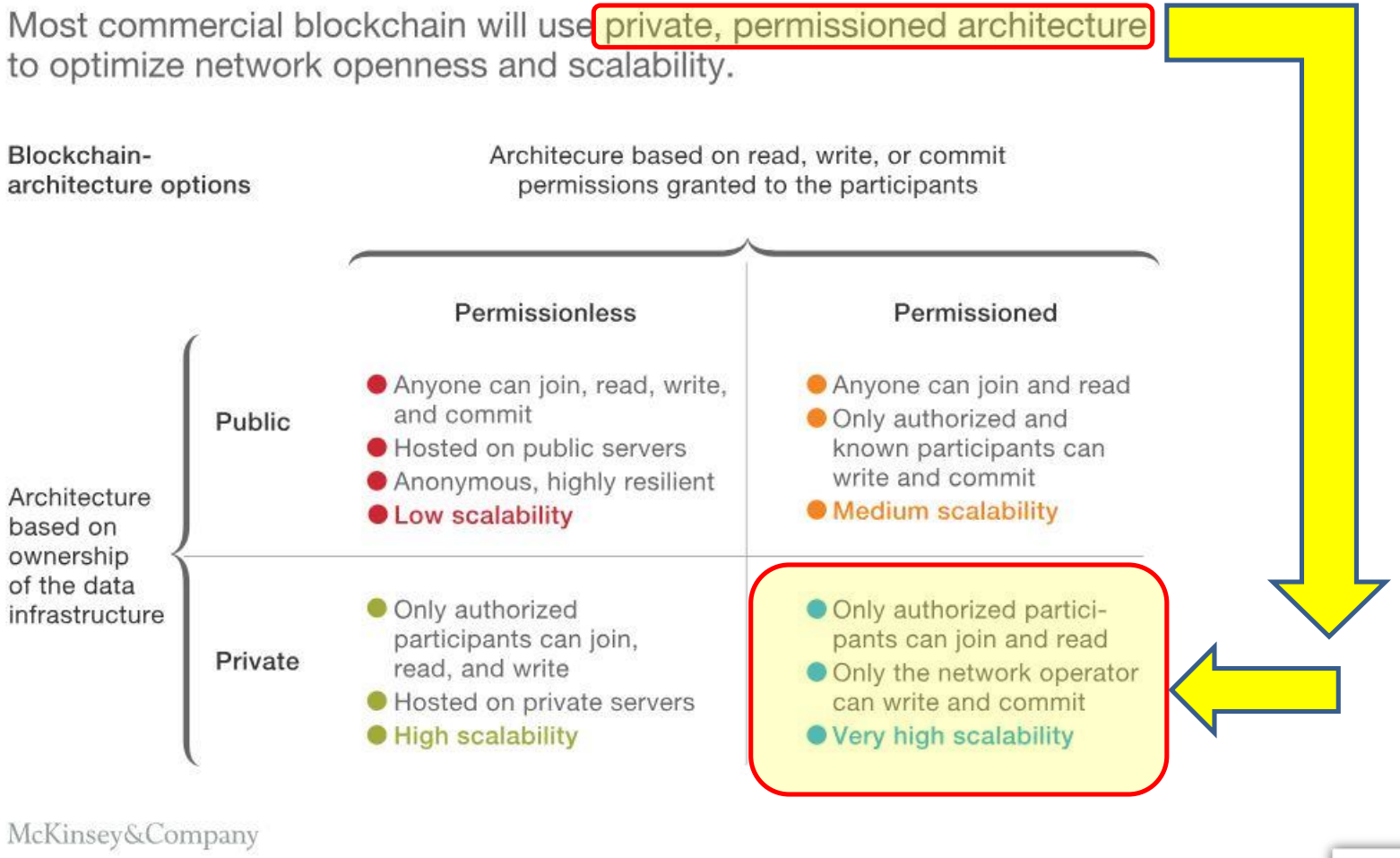
- Only safelisted (checked) participants can join the network
- Only safelisted (checked) participants can contribute computing power to the network and broadcast network data
- Access privileges determine the extent to which each safelisted participant can contribute data to the network and access data from the network

Key differences between public, permissionless blockchains and private, permissioned blockchains; **Source:** Accenture

Important Blockchain Architecture Decision

Exhibit 3

Most commercial blockchain will use private, permissioned architecture to optimize network openness and scalability.



McKinsey&Company

Other Blockchains to Explore

Other Types of Blockchains to Explore

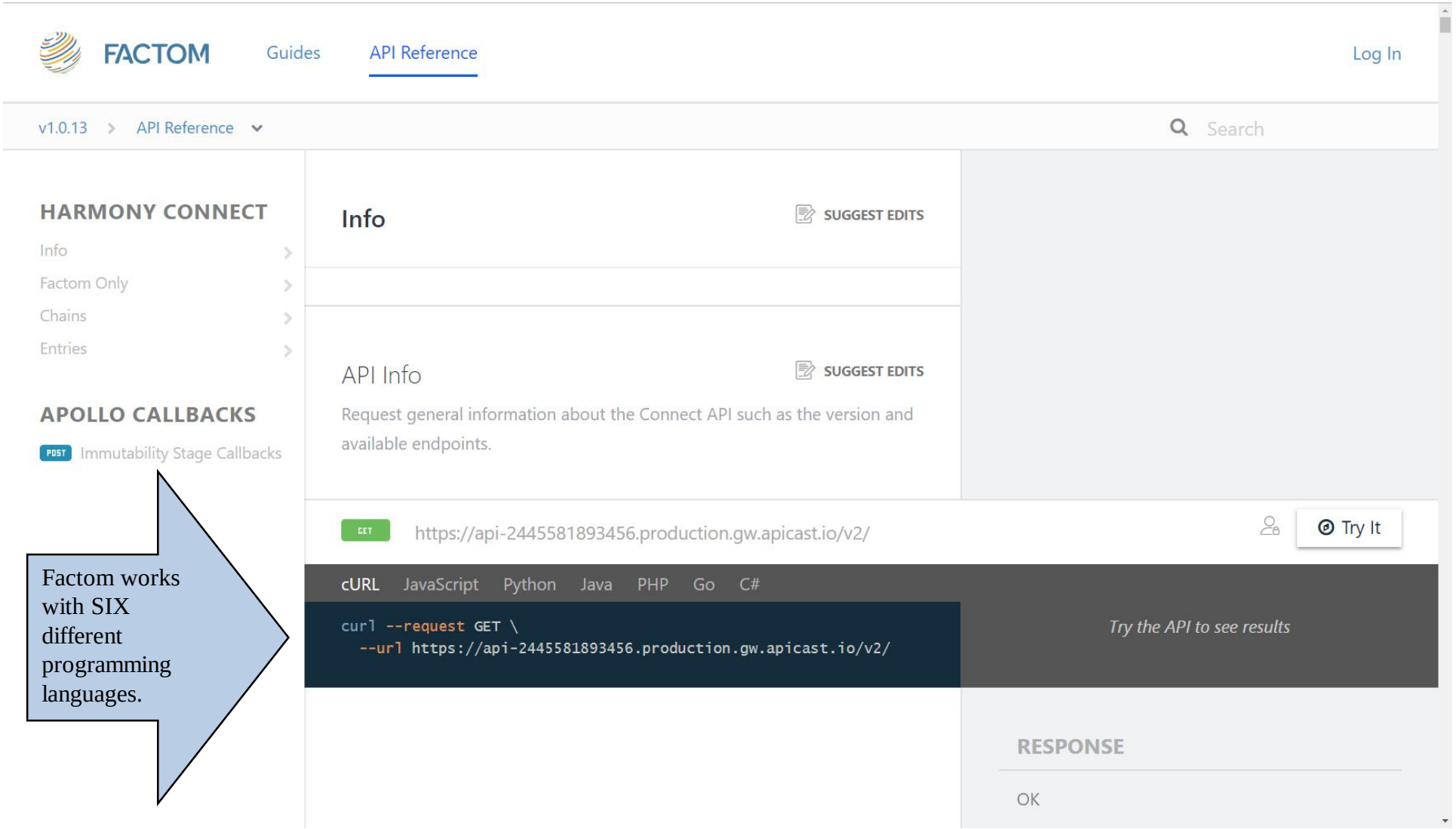
- Factom
- NEM
- BigchainDB

Factom

- Web-based
- Allows Rapid Application Development in Javascript, Python, Java, etc.
- Based in Austin, TX and in Tokyo, Japan
- http://www.factom.com
- https://apollo-docs.factom.com/

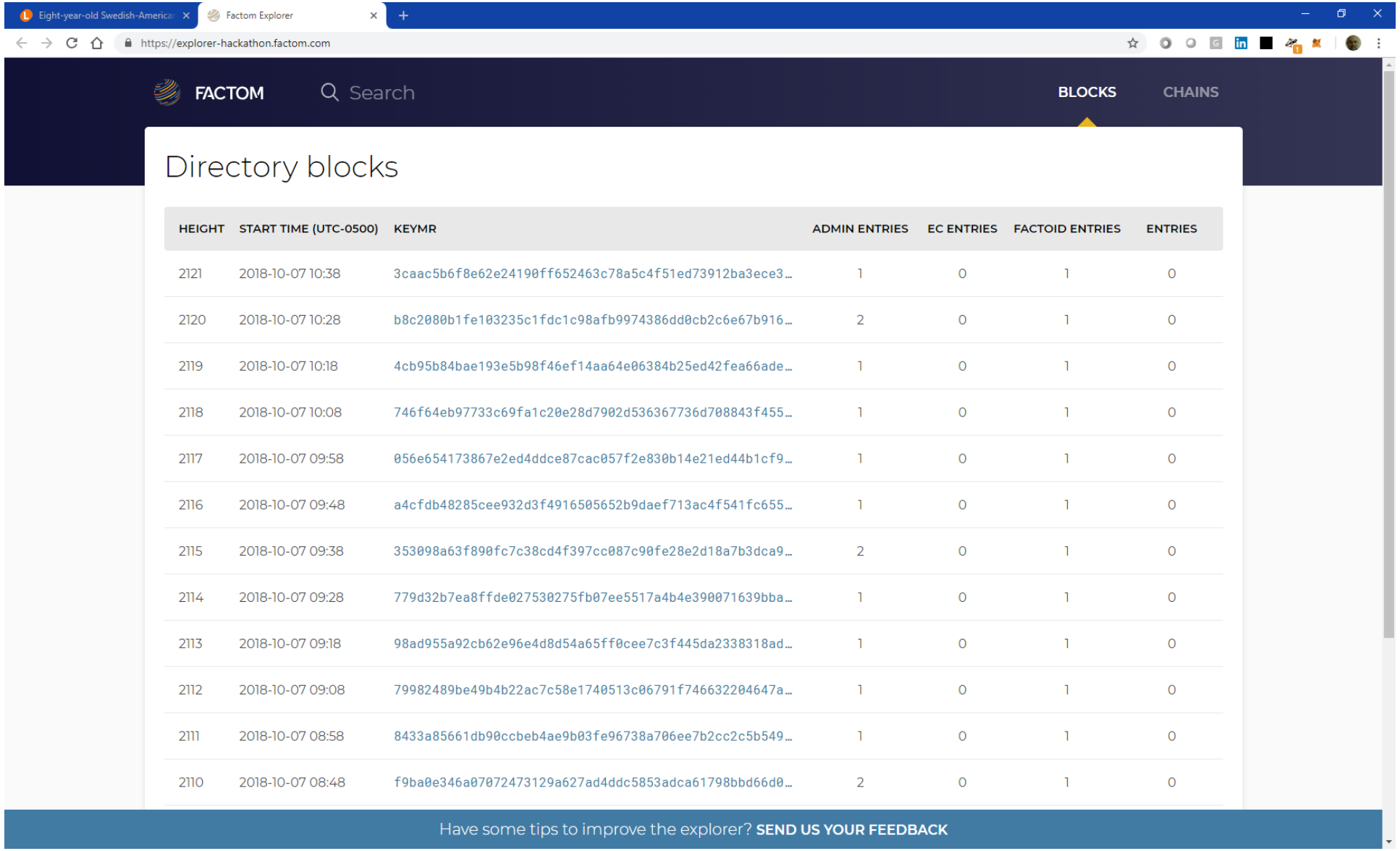
FACTOM

Factom



Factom works with SIX different programming languages.

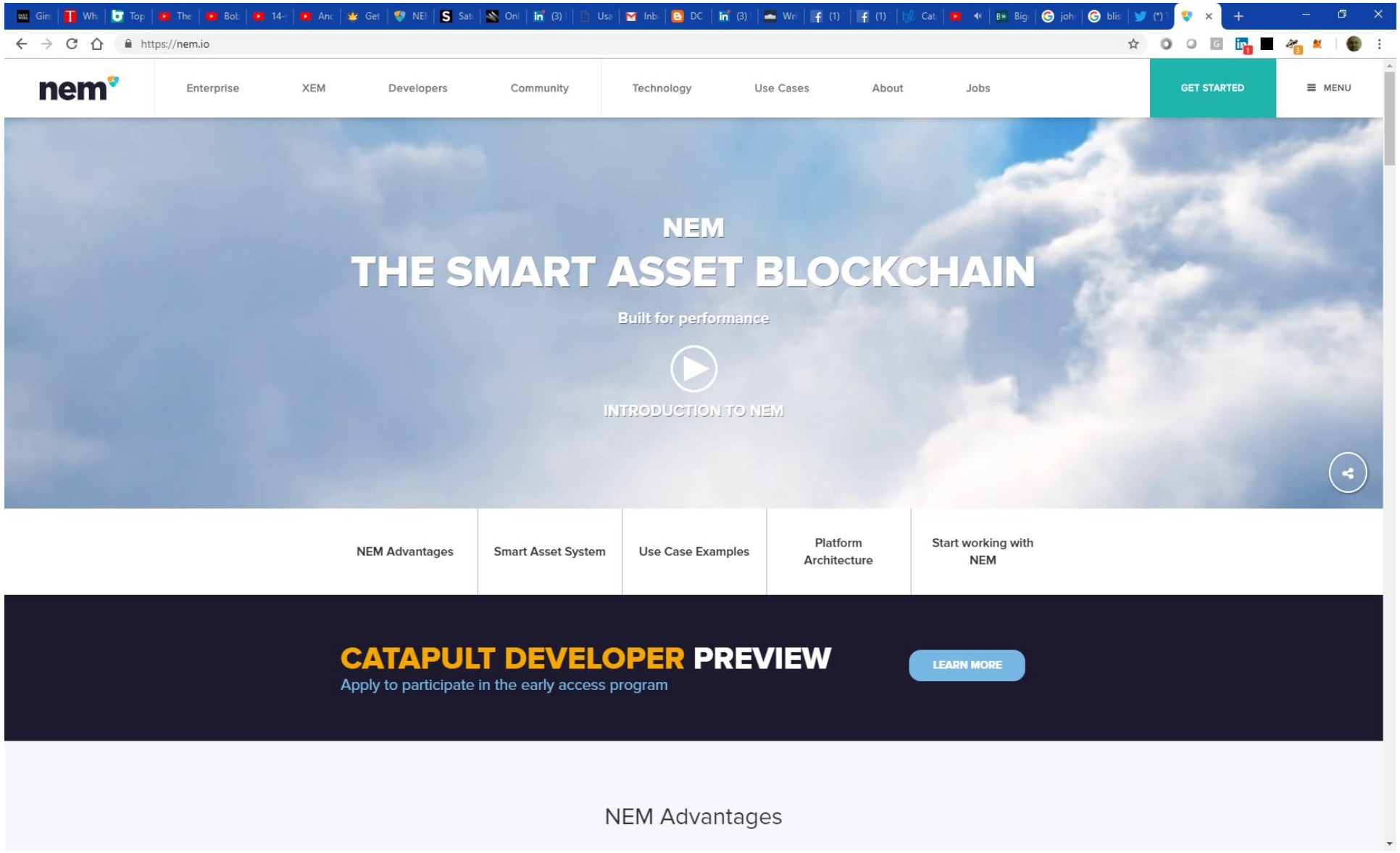
Factom



NEM

- http://nem.io
- Best case studies:
  - In America: Native American communities use to track assets
  - In Asia: NEM is used in high-performance financial applications
- Download the NEM Nano Wallet and get started

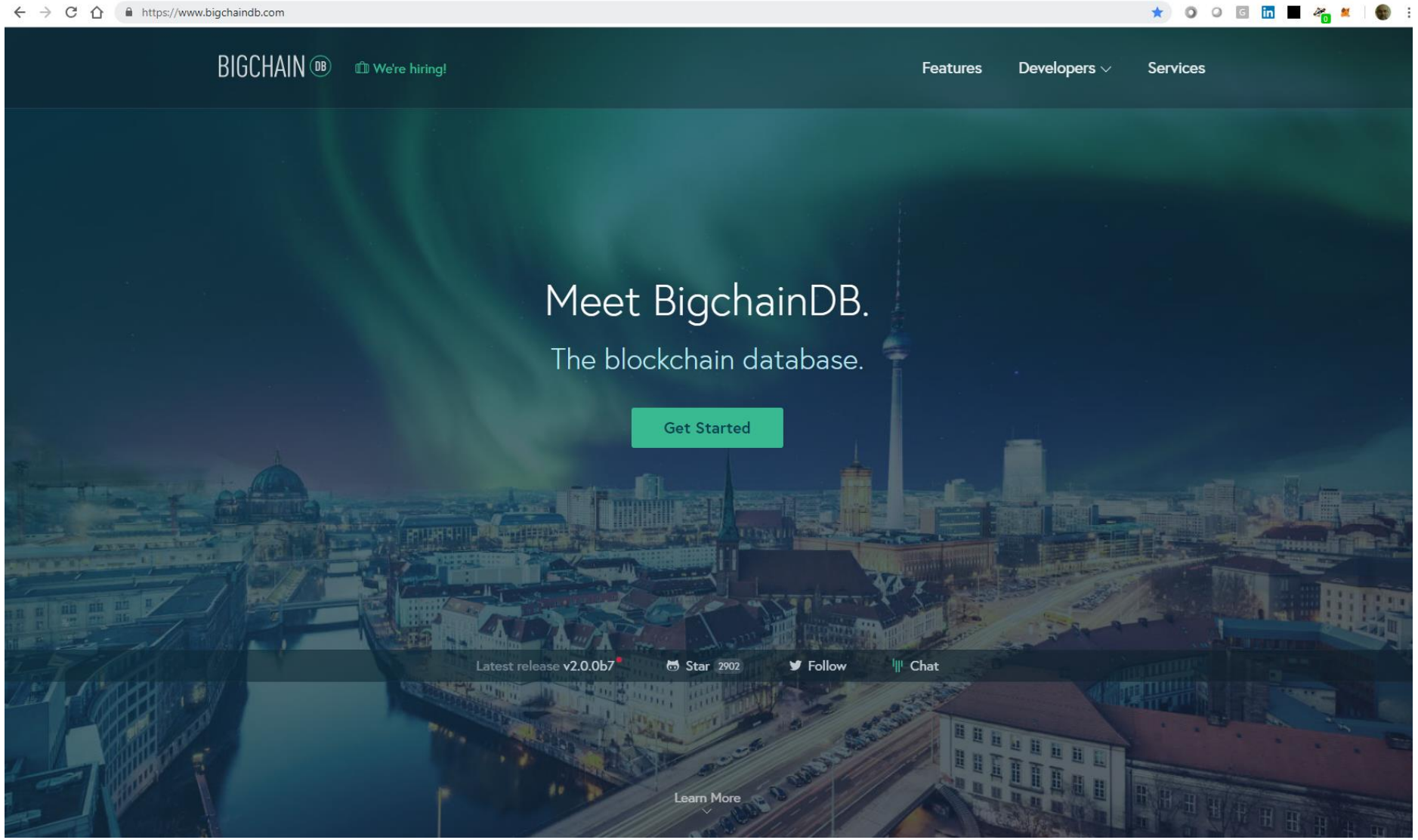
NEM



BigchainDB

- www.bigchaindb.com
- Web-based
- Demos publicly available via the web

BigchainDB



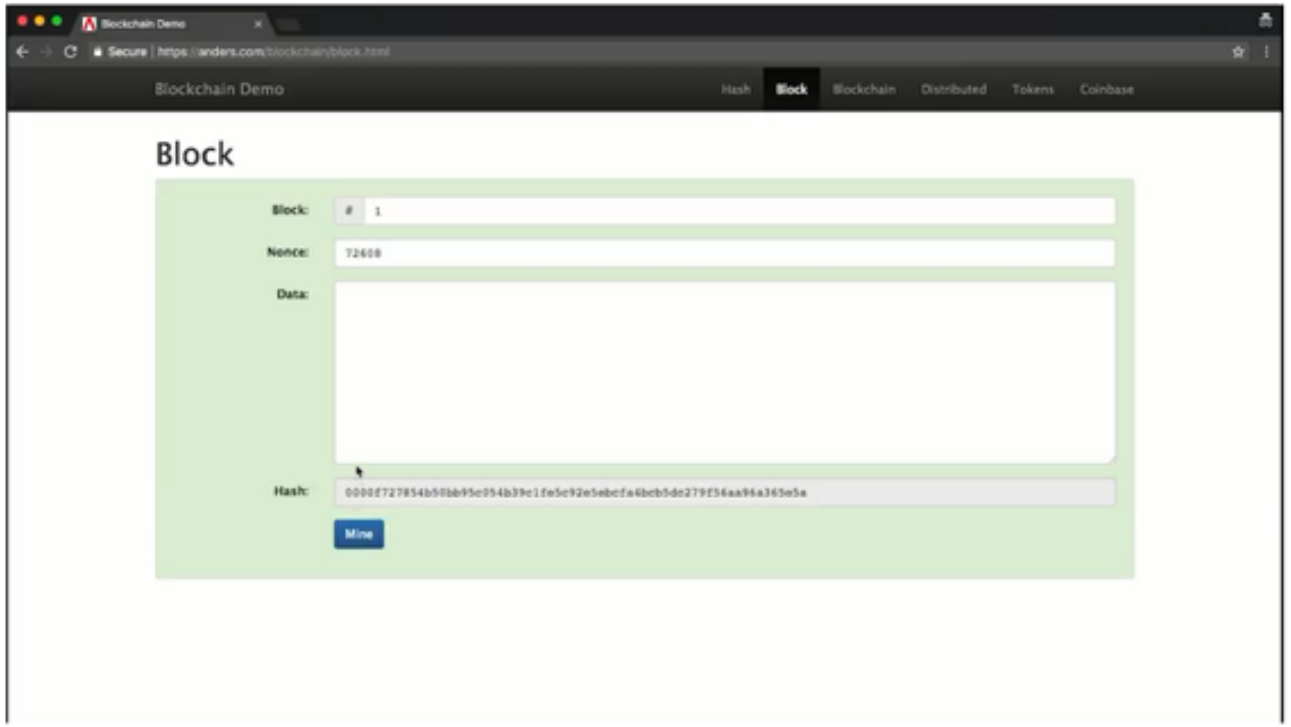
Demos from Anders.com

Blockchain – Simple Demonstration

Block Demonstration

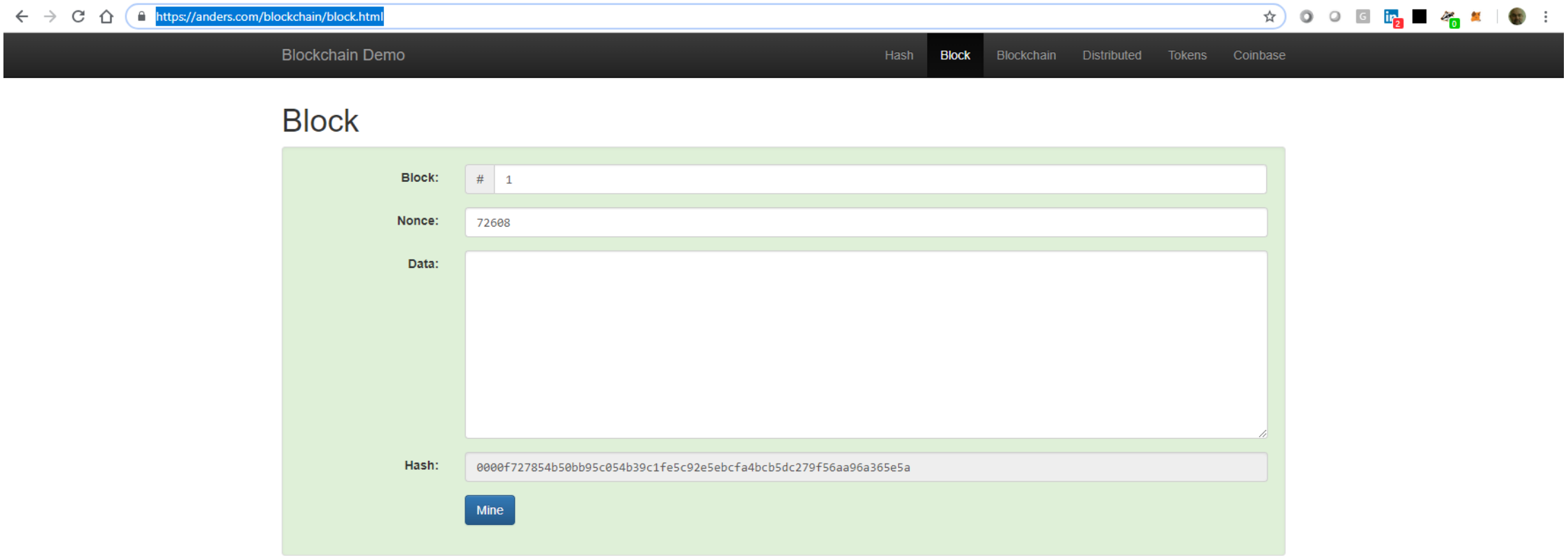
Now that you have some idea of the basics of blocks, lets go through a simple demonstration. We'll head back to the website from before to show how you can start interacting with blocks yourself.

You can follow along with this demonstration at Anders.com.



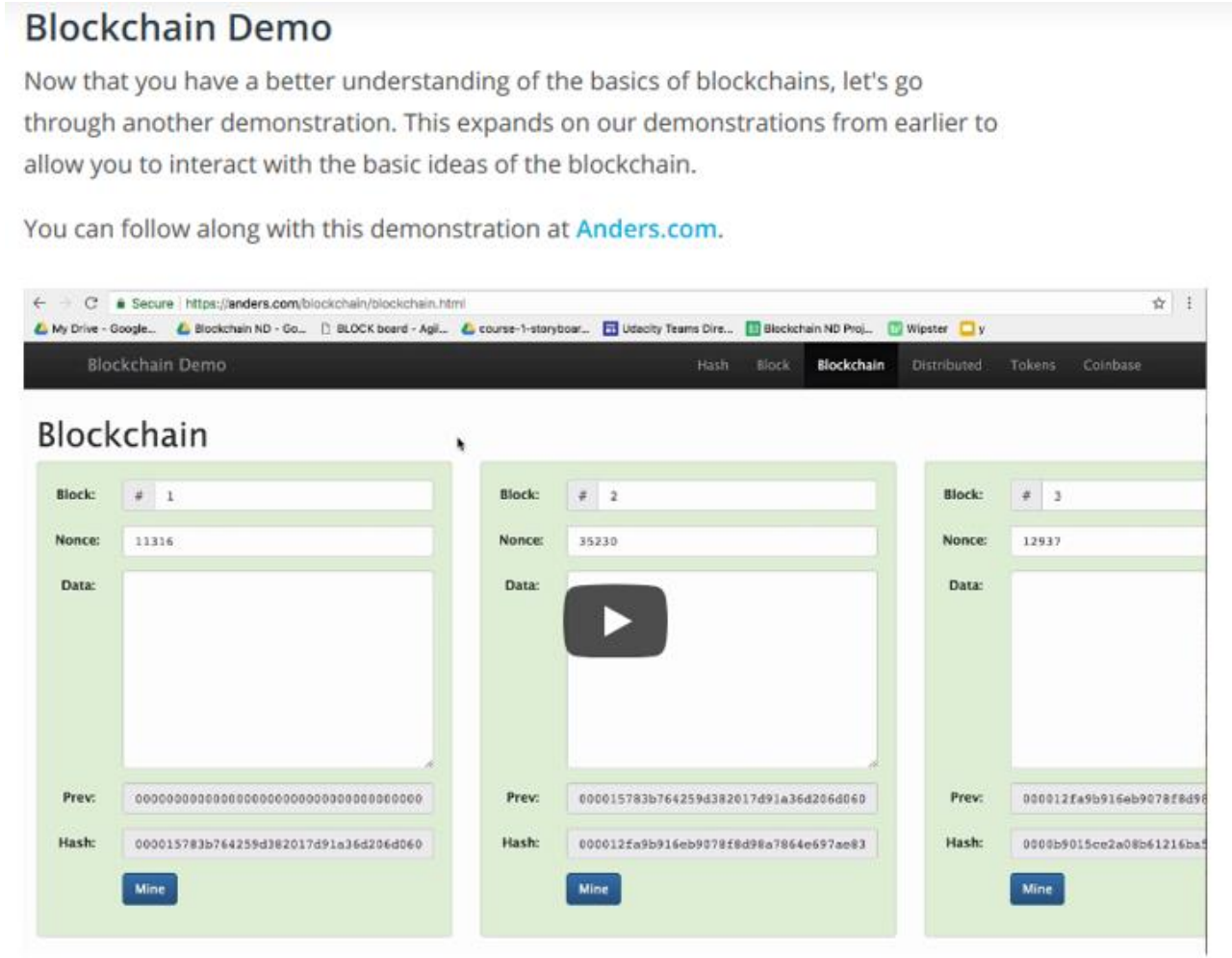
Source: Udacity Blockchain Developer Course

Blockchain – Simple Demonstration



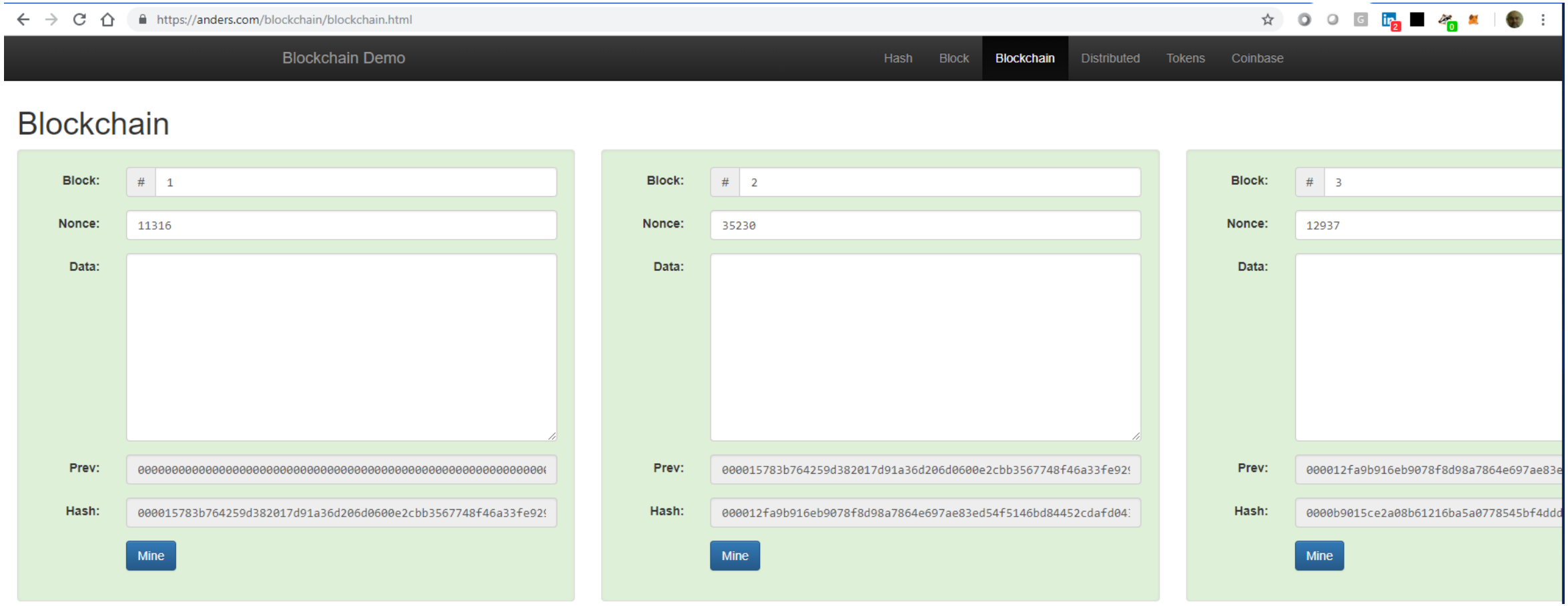
Source: https://anders.com/blockchain/block.html

Blockchain – Simple Demonstration



Source: Udacity Blockchain Developer Course

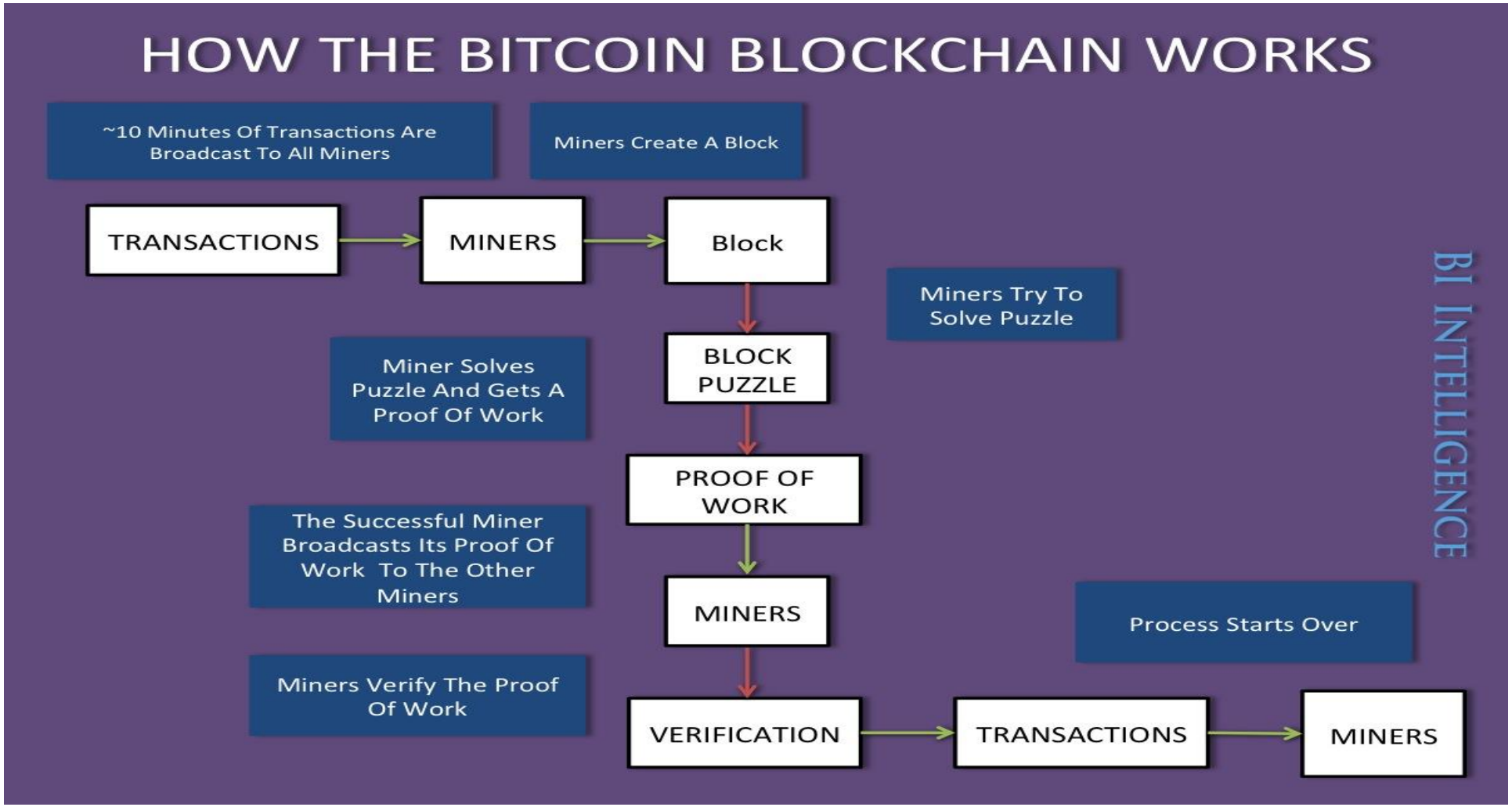
Blockchain – Simple Demonstration



Source: https://anders.com/blockchain/block.html

How Does Blockchain Work?

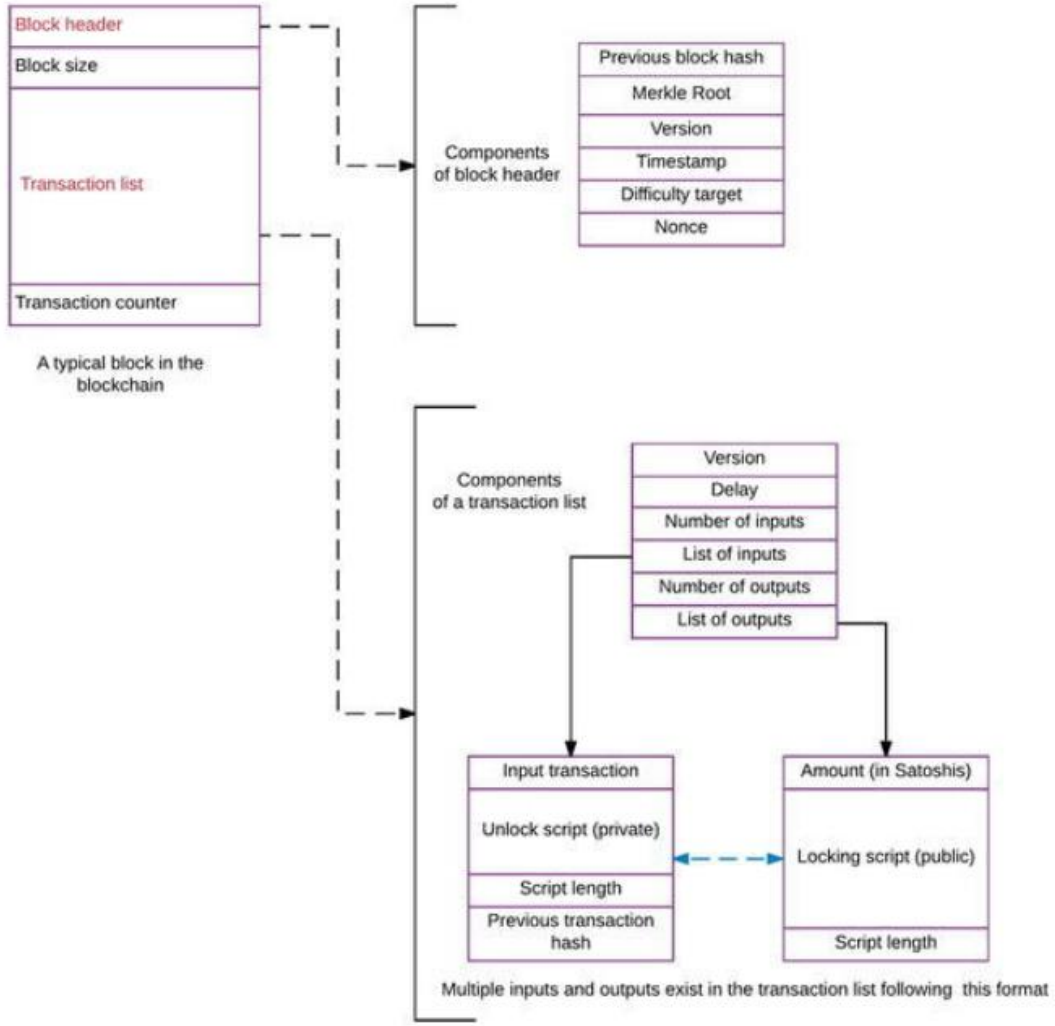
How Does Blockchain Work?



Typical Blockchain Composition

Typical Block Composition:

- Block Header
- Block Transactions



Source: **Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You** by Vikram Dhillon, David Metcalf, Max Hooper

Blockchain Mining Process

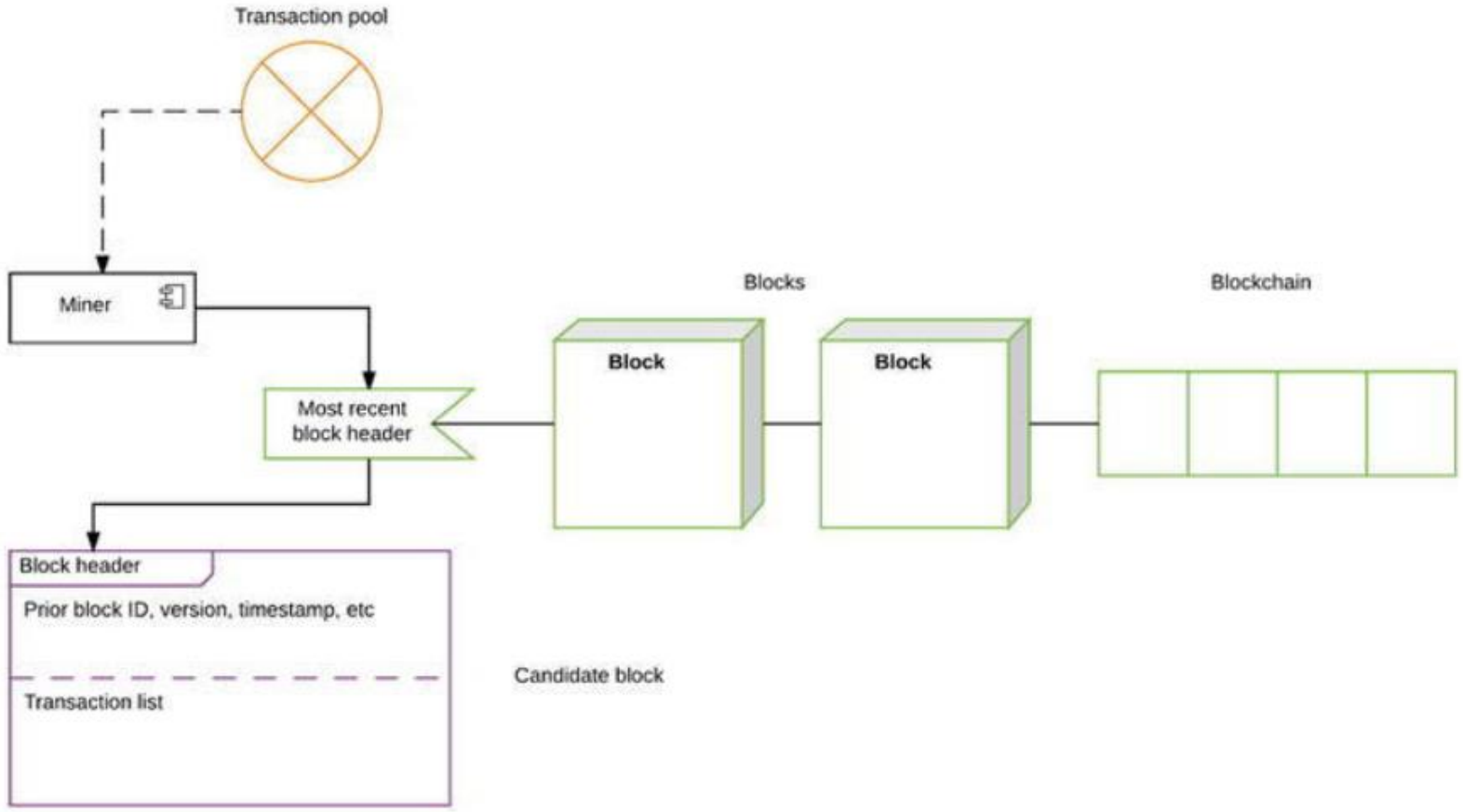


Figure 2-1.
A simplified overview of the mining process

Source: **Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You** by Vikram Dhillon, David Metcalf, Max Hooper

Bitcoin Mining Principles

1. 1.
   An increase in mining difficulty causes a decrease in the target value to compensate for the mining time.

2. 2.
   An increase in the number of miners joining the network causes an increase in the rate at which PoW is solved, decreasing the mining time. To adjust for this, mining difficulty increases and the block creation rate returns to normal.

3. 3.
   The target value is recalculated and adjusted every 2,016 blocks created, which happens in approximately two weeks.

Special Note: Many other Blockchains, including Ethereum, apply these same principles.

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

Bitcoin Mining Principles

Note

The term *mining* is used because the process is similar to the mining of rare metals. It is very resource intensive and it makes new currency avaliable at a slow rate, just like the miners in the Bitcoin protocol getting rewarded.

allows it to be very resilient. Miners are the heartbeat of the Bitcoin network and they have two main incentives for participation:

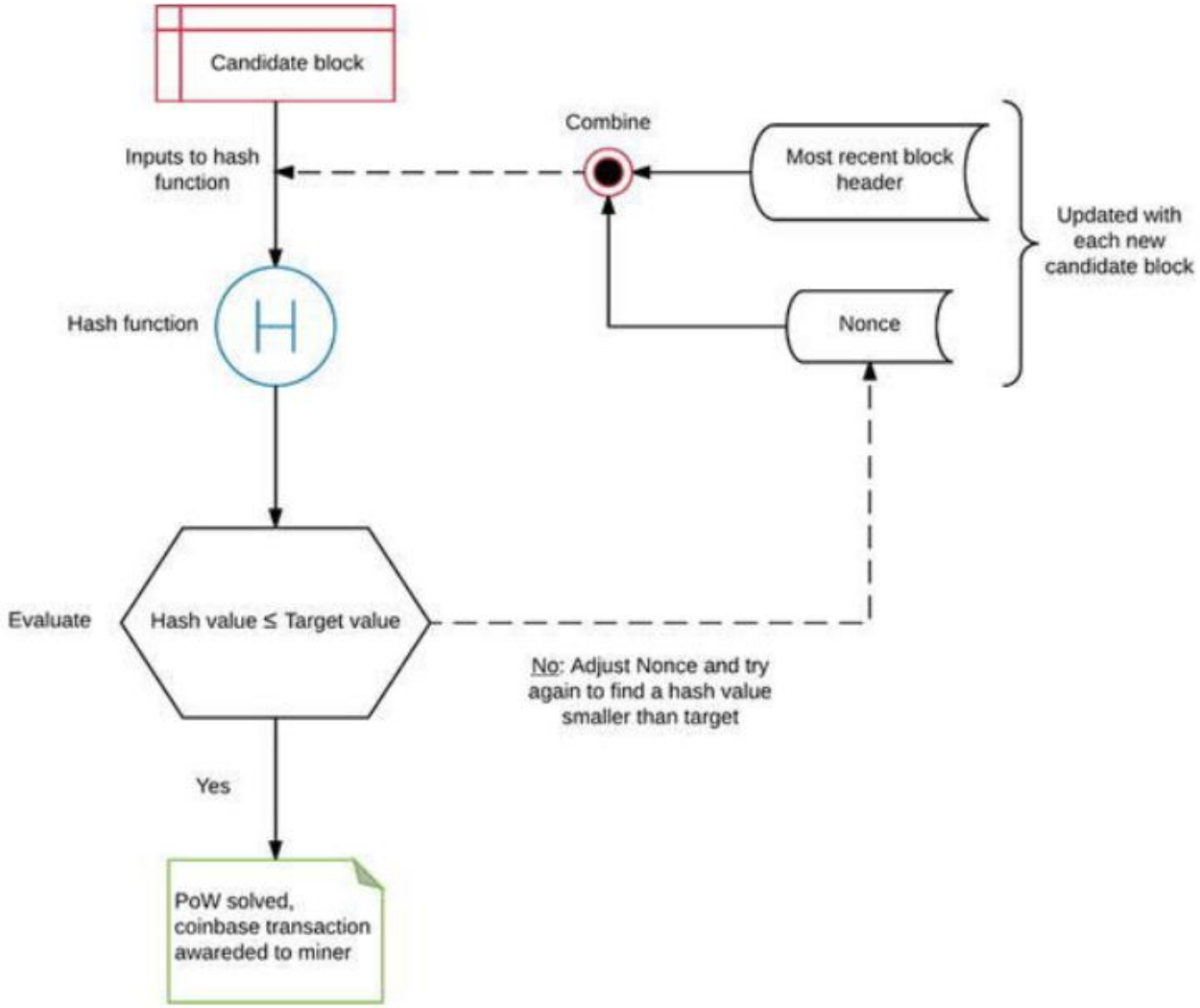
- **The first transaction to be packaged in a block is called the coinbase transaction. This transaction is the reward that the winning miner receives after mining the block and announcing it on the network.**
- **The second reward comes in the form a fee charged to the users of the network for sending transactions. The fee is given to the miners for including the transactions in a block. This fee can also be considered a miner's income because as more and more Bitcoins are mined, this fee will become a significant portion of the income.**

Special Note: Many other Blockchains, including Ethereum, apply these same principles.

Source: Drescher, D. (2

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

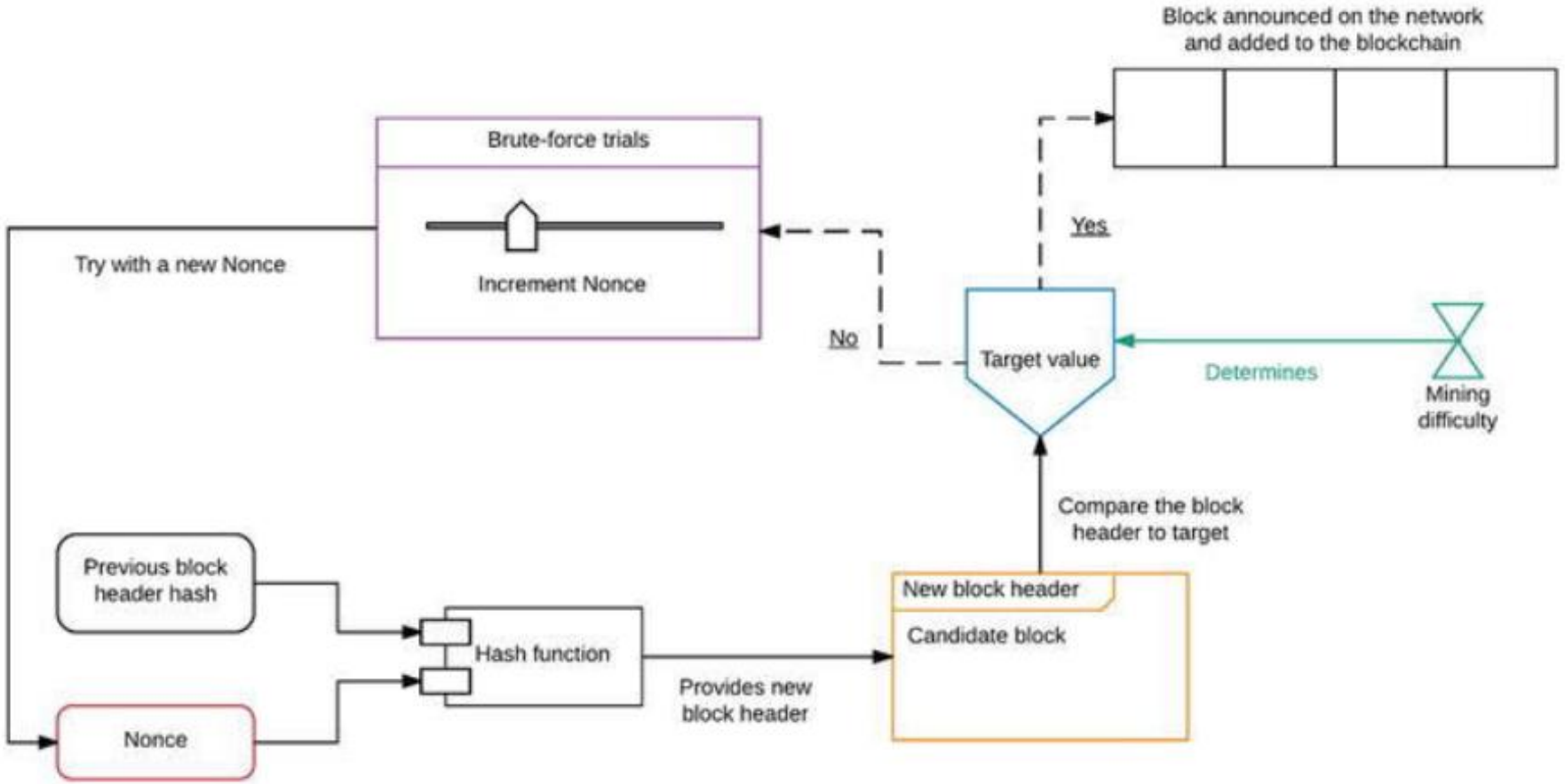
Mining Principles – Proof of Work



Special Note: Many other Blockchains, including Ethereum, apply these same principles.

Source: **Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You** by Vikram Dhillon, David Metcalf, Max Hooper

Mining Principles – Solving the Proof of Work



Special Note: Many other Blockchains, including Ethereum, apply these same principles.

Source: **Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You** by Vikram Dhillon, David Metcalf, Max Hooper

Mining Principles – Block Creation

1. Get the root of the Merkle tree that contains the transaction data to be added.
2. Create a hash reference to the header of that block that will be the predecessor from the new block header's point of view.
3. Obtain the required difficulty level.
4. Get the current time.
5. Create a preliminary block header that contains the data mentioned in points 1 to 4.
6. Solve the hash puzzle for the preliminary block header.
7. Finish the new block by adding the nonce that solves the hash puzzle to the preliminary header.

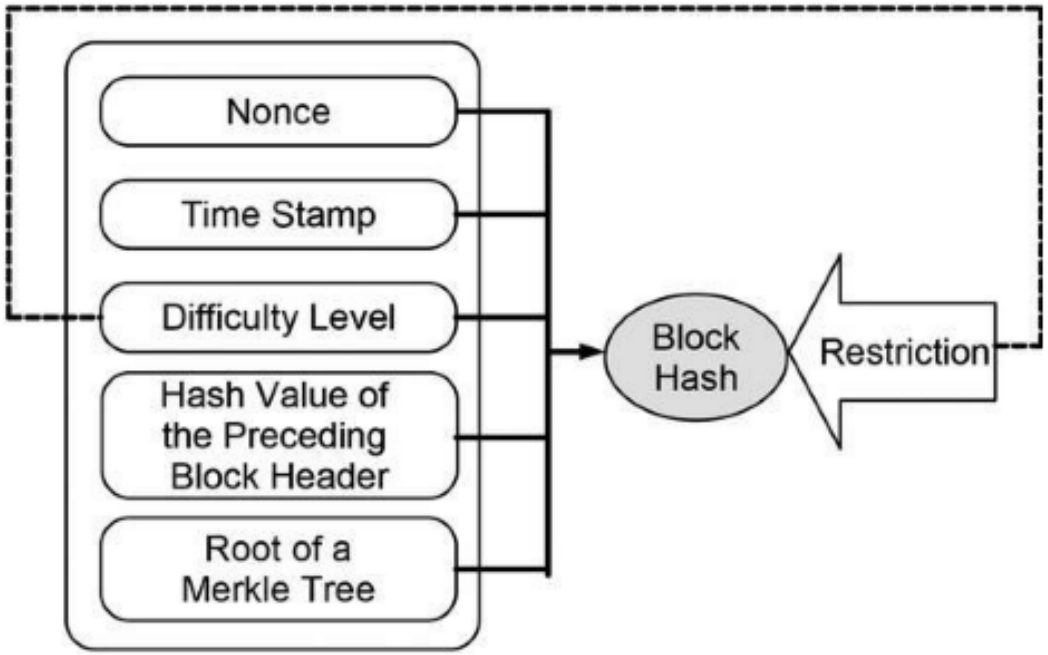


Figure 16-1. *Schematic illustration of the hash puzzle required to be solved when adding a new block to the blockchain-data- structure*

Special Note: Many other Blockchains, including Ethereum, apply these same principles.

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

How Blockchain Block Creation Works – In Detail – Part 1

The procedure that governs how nodes deal with new transaction data and blocks they receive from their peers consists of the following rules (the rules printed in bold are the one that establish the two-step rhythm):

1. New transaction data as well as new blocks are forwarded to all nodes in a gossip fashion.
2. Each node collects new transaction data in an inbox and selects them for processing.
3. **Each node processes new blocks immediately with highest priority.**

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

How Blockchain Block Creation Works – In Detail – Part 2

4. Each node processes new transaction data by validating them for authorization and formal and semantic correctness.
5. Each node collects only valid transaction data into a Merkle tree and starts creating a new block by solving its hash puzzle.
6. **As soon as a node finishes the hash puzzle, it sends the newly created block to all other nodes.**
7. Each node processes new blocks by verifying the solution of its hash puzzle and by verifying all its containing transaction data for formal correctness, semantic correctness, and authorization.

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

How Blockchain Block Creation Works – In Detail – Part 3

8. Each node adds valid blocks to its own copy of the blockchain-data-structure.
9. If a newly arrived block has been identified as invalid, it will be discarded and the nodes continue with processing transaction data or with finishing the hash puzzle of a new block.
10. If a newly arrived block has been identified as valid, the node removes those transactions that are contained in the new block from its own inbox and starts with processing transaction data and the creation of a new block.

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

How Blockchain Block Creation Works – In Detail – Part 4

11. If a block that was added to the blockchain-data-structure is identified as invalid or useless later on, that block as well as all its subsequent blocks will be removed[2] from the blockchain-data-structure and their transactions will be added to the inbox to be processed again.
12. The node whose block was accepted will receive the fees for all transactions contained in the block as reward.
13. If a block is removed from the blockchain-data-structure, then the reward for adding it is withdrawn from the node that initially received it.

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

WHY Blockchain Block Creation Works – In Detail – Part 1

The reasons the preceding rules work are:

- Due to rule 1, all nodes receive all information needed to validate and add transaction data.
- Due to rule 2, nodes process new transaction data they receive.
- Due to rule 3, the blocks created by other nodes are processed immediately on arrival at the nodes inbox.
- Due to rule 4, only valid transaction data are added to the blockchain-data-structure

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

WHY Blockchain Block Creation Works – In Detail – Part 2

- Due to rule 5, all nodes take part in a race for solving the hash puzzle. Due to the nature of the hash puzzle it is unpredictable which node will solve it first.
- Due to rule 6, all nodes are informed when a node solves the hash puzzle of a new block.
- Due to rules 6 and 3, all nodes receive the newly created block and recognize the winner of the race for solving the hash puzzle.
- Due to rule 7, all nodes of the system review and verify newly created blocks and ensure that only correct blocks are accepted.

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

WHY Blockchain Block Creation Works – In Detail – Part 3

- Due to rule 8, all nodes add new blocks to their own copy of the blockchain-data-structure and hence grow the transaction history.
- Due to rule 9, the collectively maintained transaction history is kept free of invalid transactions and hence maintains integrity.
- Due to rule 10, no transaction data will be added twice.
- Due to rule 11, no valid transaction will get lost even if previously processed blocks are reprocessed.

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

WHY Blockchain Block Creation Works – In Detail – Part 4

- Due to rule 11, the system is able to perform ex post validity checks on the transaction history and correct it retrospectively.
- Due to rule 12, nodes have an incentive to process transactions and to create new blocks quickly.
- Due to rule 12, all nodes have an incentive to inform all other nodes about a new block because earning a reward depends on having transactions examined and accepted by all other nodes.
- Due to rule 13, nodes have an incentive to work correctly, to avoid accepting any invalid transaction data, or producing invalid blocks.
- Due to rule 13, nodes have an incentive to review and revalidate blocks and transactions in a retrospective way.

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

Break

Topic 5: Ethereum Blockchain Technology

Overview of Ethereum

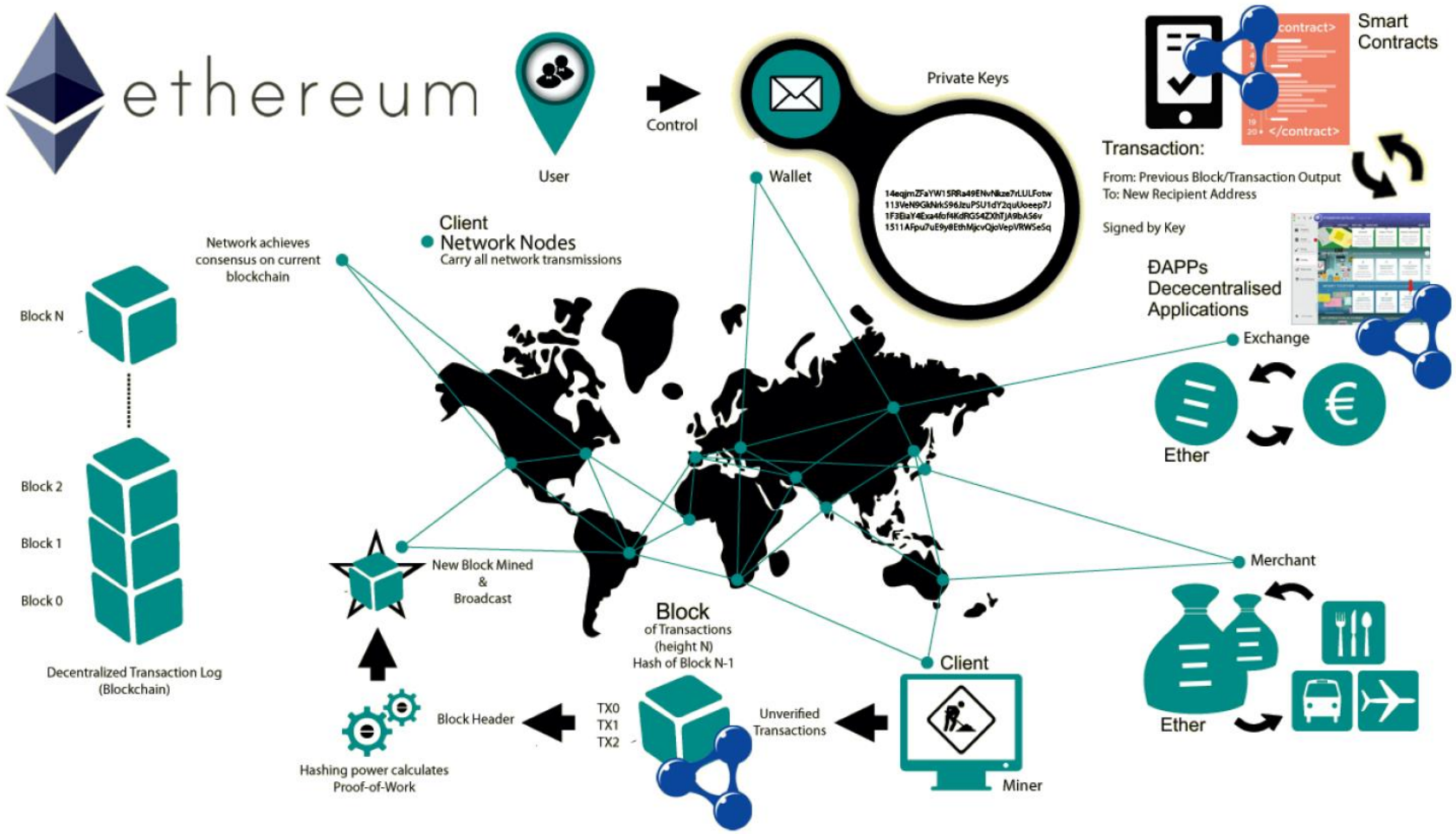


Fig. 6. Ethereum framework elements, modified from [39, p.16]

Source: https://www.researchgate.net/publication/315619465_A_more_pragmatic_Web_30_Linked_Blockchain_Data

Ethereum Public Blockchain

- Ethereum was developed initially for public chain deployment, where trustless transaction requirements outweigh absolute performance. The current public chain consensus algorithms (notably PoW) are overkill for networks with trusted actors and high throughput requirements.
- Public chains by definition have limited (at least initially) privacy and permissioning requirements. Although Ethereum does enable permissioning to be implemented within the smart contract and network layers, it is not readily compatible out of the box with traditional enterprise security and identity architectures or data privacy requirements.
- Naturally, the current Ethereum improvement process (dominated by Ethereum improvement proposals) is largely dominated by public chain matters, and it has been previously challenging for enterprise IT requirements to be clarified and prioritized within it.

Publicly released on July 30, 2015

Source: **Blockchain Basics: A Non-technical Introduction in 25 Steps**
by Daniel Drescher

Ethereum Overview – Part 1

Ethereum is a decentralized platform, which allows us to deploy DApps on top of it. Smart contracts are written using the solidity programming language. DApps are created using one or more smart contracts. Smart contracts are programs that run exactly as programmed without any possibility of downtime, censorship, fraud, or third party interface. In Ethereum, smart contracts can

Source: Building Blockchain Projects, by Narayan Prusty (Published by Packt.)

Ethereum Overview – Part 2

be written in several programming languages, including Solidity, LLL, and Serpent. Solidity is the most popular of those languages. Ethereum has an internal currency called ether. To deploy smart contracts or to call their methods, we need ether. There can be multiple instances of a smart contract just like any other DApp, and each instance is identified by its unique address. Both user accounts and smart contracts can hold ether.

Ethereum uses blockchain data structure and proof-of-work consensus protocol. A method of a smart contract can be invoked via a transaction or via another method. There are two kinds of nodes in the network: regular nodes and miners. Regular nodes are the ones that just have a copy of the blockchain, whereas miners build the blockchain by mining blocks

Source: Building Blockchain Projects, by Narayan Prusty (Published by Packt.)

Ethereum Consensus Overview – Part 1

Every node in the Ethereum network holds a copy of the blockchain. We need to make sure that nodes cannot tamper with the blockchain, and we also need a mechanism to check whether a block is valid or not. And also, if we encounter two different valid blockchains, we need to have a way to find out which one to choose.

Ethereum uses the proof-of-work consensus protocol to keep the blockchain tamper-proof. A proof-of-work system involves solving a complex

Source: Building Blockchain Projects, by Narayan Prusty (Published by Packt.)

Ethereum Consensus Overview – Part 2

puzzle to create a new block. Solving the puzzle should require a significant amount of computational power thereby making it difficult to create blocks. The process of creating blocks in the proof-of-work system is called mining. Miners are the nodes in the network that mine blocks. All the DApps that use proof-of-work do not implement exactly the same set of algorithms. They may differ in terms of what the puzzle miners need to solve, how difficult the puzzle is, how much time it takes to solve it, and so on. We will learn about proof-of-work with respect to Ethereum.

Anyone can become a miner in the network. Every miner solves the puzzle individually; the first miner to solve the puzzle is the winner and is rewarded with five ether and transaction fees

Source: Building Blockchain Projects, by Narayan Prusty (Published by Packt.)

Ethereum Consensus Overview – Part 3

of all the transactions in that block. If you have a more powerful processor than any other node in the network, that doesn't mean that you will always succeed because the parameters for the puzzle are not exactly same for all the miners. But instead, if you have a more powerful processor than any other node in the network, it gives you a higher chance at succeeding. Proof-of-work behaves like a lottery system, and processing power can be thought as the number of lottery tickets a person has. Networks security is not measured by total number of miners; instead, it's measured by the total processing power of the network.

There is no limit to the number of blocks the blockchain can have, and there is no limit to the total ether that can be produced. Once a miner

Source: Building Blockchain Projects, by Narayan Prusty (Published by Packt.)

Ethereum Consensus Overview – Part 4

successfully mines a block, it broadcasts the block to all other nodes in the network. A block has a header and a set of transactions. Every block holds hash of the previous block, thereby creating a connected chain.

Let's see what the puzzle the miners need to solve is and how it's solved at a high level. To mine a block, first of all, a miner collects the new un-mined transactions broadcasted to it, and then it filters out the not-valid transactions. A trans-action to be valid must be properly signed using the private key, the account must have enough balance to make the transaction, and so on. Now the miner creates a block, which has a header and content. Content is the list of transactions that the block contains. The header contains things

Source: Building Blockchain Projects, by Narayan Prusty (Published by Packt.)

Ethereum Consensus Overview – Part 5

such as the hash of the previous block, block number, nonce, target, timestamp, difficulty, address of the miner, and so on. The timestamp represents the time at the block's inception. Then nonce is a meaningless value, which is adjusted in order to find the solution to the puzzle. The puzzle is basically to find such nonce values with which when the block is hashed, the hash is less than or equal to the target. Ethereum uses ethash hashing algorithm. The only way to find the nonce is to enumerate all possibilities. The target is a 256-bit number, which is calculated based on various factors. The difficulty value in the header is a different representation of the target to make it easier to deal with. The lower the target, the more time it takes to find the nonce, and the higher the target, the less time it takes to find the nonce.

Source: Building Blockchain Projects, by Narayan Prusty (Published by Packt.)

Ethereum Consensus Overview – Part 6

Here is the formula to calculate the difficulty of the puzzle:

```
current_block_difficulty = previous_block_difficulty + previous_block_difficulty // 2048 * max(1 - (current_block_timestamp - previous_blocktimestamp) // 10, -99) + int(2 ** ((current_block_number // 100000) - 2))
```

Now any node in the network can check whether the blockchain they have is valid or not by first checking whether the transactions in the blockchain are valid, the timestamp validation, then whether the target and nonce of all the blocks are valid, a miner has assigned a valid reward itself, and so on.

Ethereum Consensus Overview – Part 7

If a node in the network receives two different valid blockchains, then the blockchain whose combined difficulty of all blocks is higher is considered to be the valid blockchain.

Now, for example, if a node in the network alters some transactions in a block, then the node needs to calculate the nonce of all the succeeding blocks. By the time it re-finds the nonce of the succeeding blocks, the network would have mined many more blocks and therefore reject this blockchain as its combined difficulty would be lower.

Source: Building Blockchain Projects, by Narayan Prusty (Published by Packt.)

Ethereum Blockchain Validator Algorithm

1. Check if the previous block referenced exists and is valid.
2. Check that the timestamp of the block is greater than that of the referenced previous block and less than 15 minutes into the future.
3. Check that the block number, difficulty, transaction root, uncle root and gas limit (various low-level Ethereum-specific concepts) are valid.
4. Check that the nonce on the block is valid, showing the evidence of proof of work.
5. Apply all transactions in this now-validated block to the EVM state. If any errors are thrown, or if total gas exceeds the GASLIMIT, return an error and roll back the state change.
6. Add the block reward to the final state change.
7. Check that the Merkle tree root final state is equal to the final state root in the block header.

Source: Introducing Ethereum and Solidity – by Chris Dannen (Published by Apress)

Merkle Trees

Thanks to the block header, it's quick and easy for a node to look for, read, or verify block data. In Bitcoin, the block header is an 80-byte chunk of data that includes the Merkle root as well as five other things. The Bitcoin block header contains:

- A hash of the previous block header
- A timestamp
- A mining difficulty value
- A proof-of-work nonce
- A root hash for the Merkle tree containing the transactions for that block

Merkle trees are ideal for storing transaction ledgers, but that's about it. From the perspective of the EVM, one limitation of the Merkle tree is that although it can prove or disprove the inclusion of transactions in the root hash, it can't prove or query the current state of the network, such as a given user's account holdings.

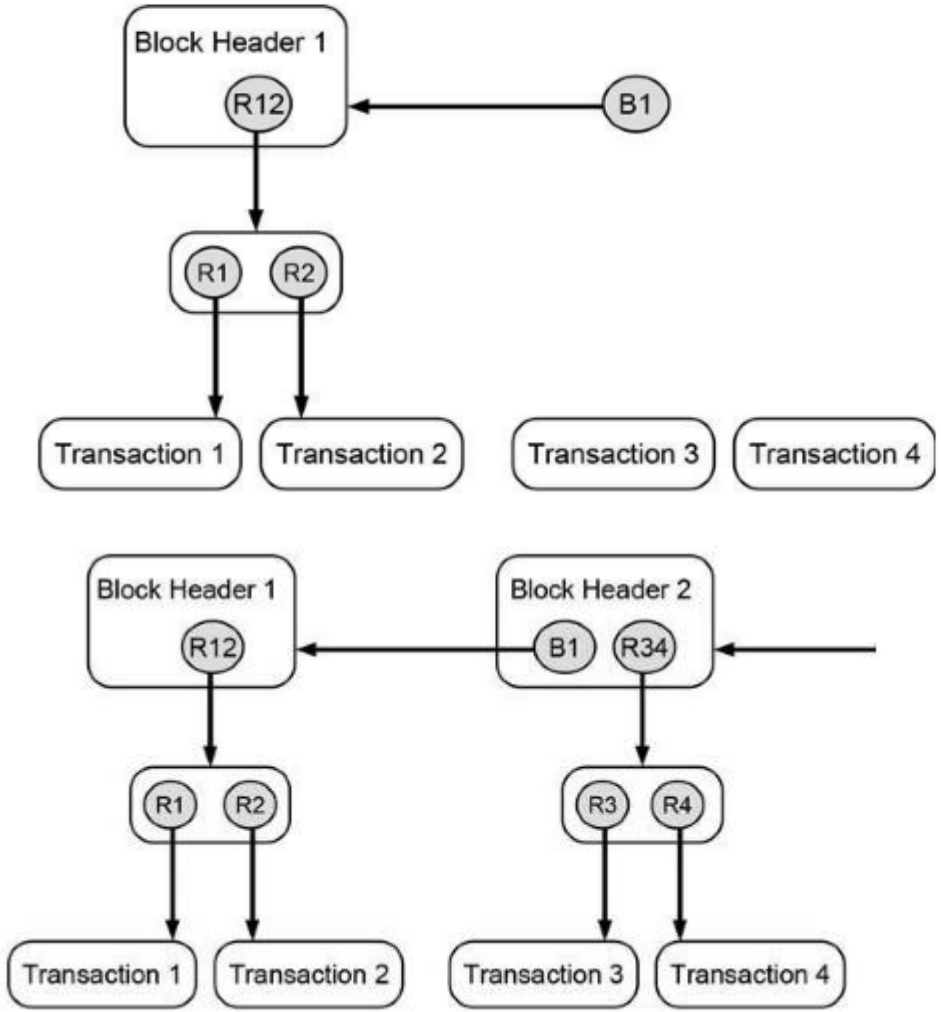
Source: Introducing Ethereum and Solidity – by Chris Dannen (Published by Apress)

Special Note:

Vitalik Buterin, the inventor of Ethereum calls Merkle Trees, "Merkle Tries"

Merkle Trees

- Merkle Trees are used to add transactions to Blocks in Bitcoin Blockchains

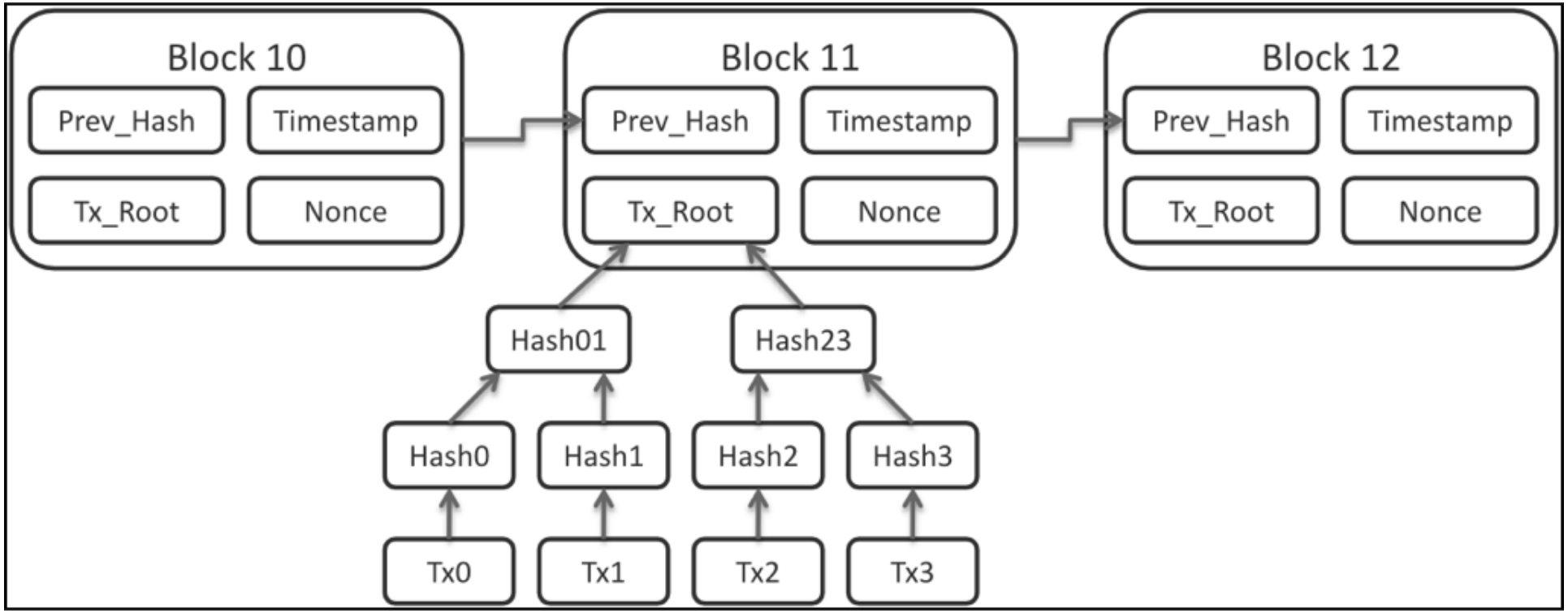


Special Note:

The inventor of Ethereum calls Merkle Trees, “Merkle Tries”

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

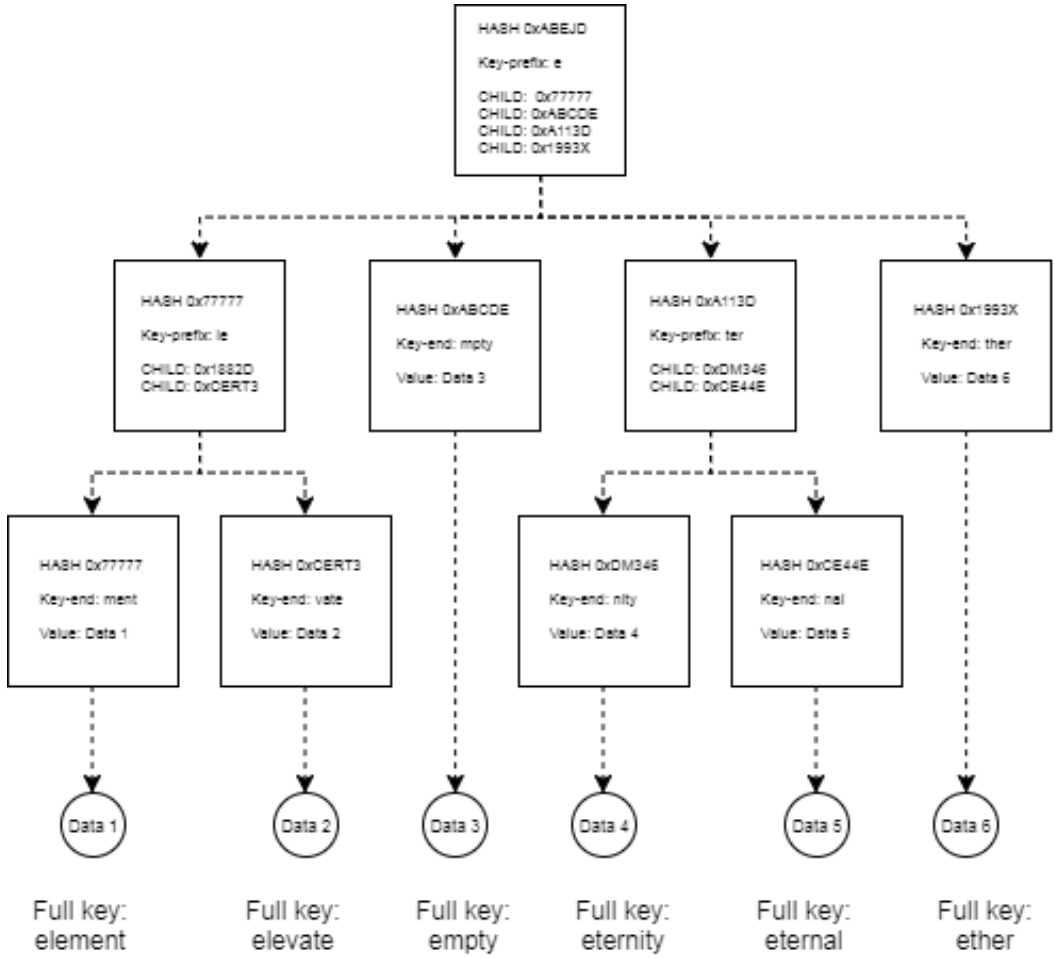
Merkle Trees



Source: Nakamoto, S. (2008).

Merkle Trees

- Merkle Patricia Trees (MPT) data structures are used to add transactions to Blocks in Ethereum Blockchains to permit the use of Smart Contracts
- MPTs use private and public keys to authenticate
- The Ethereum Blockchain is categorized as "Turing Complete" because it can be programmed using languages, like Solidity and Java, and Javascript that contain looping and testing capabilities.



Source: Peterson, O. (2018). An Introduction of Programmable Smart Contracts in Ethereum (Pt 1). Retrieved from https://www.linkedin.com/pulse/introduction-programmable-smart-contracts-ethereum-p1-%CE%BE%CE%BE%CE%BE-oliver/

Merkle Patrica Trees

To remedy this shortcoming and allow the EVM to run stateful contracts, every block header in Ethereum contains not just one Merkle (transaction) tree, but *three* trees for three kinds of objects:

Transaction tree

Receipts tree (data showing the outcome of each transaction)

State tree

To make this possible, the Ethereum protocol combines the Merkle tree with the other tree structure we described above, the Patricia tree. This tree structure is fully deterministic: two Patricia trees with the same (key/value) bindings will always have the same root hash, providing increased efficiency for common database operations such as inserts, lookups, and deletes.[12] It is therefore possible for Ethereum clients to get verifiable answers to all sorts of queries it makes to the network, such as the following:

Has transaction *X* been included in block? (Handled by the transaction tree.)

Tell me all instances of event *Y* in the last 30 days. (Handled by the receipts tree.)

What is the current balance of contract account *Z*? (Handled by the state tree.)

work and why they were chosen, check out http://trees.eth.guide .

Source: Introducing Ethereum and Solidity – by Chris Dannen (Published by Apress)

High-Level DApp Architecture

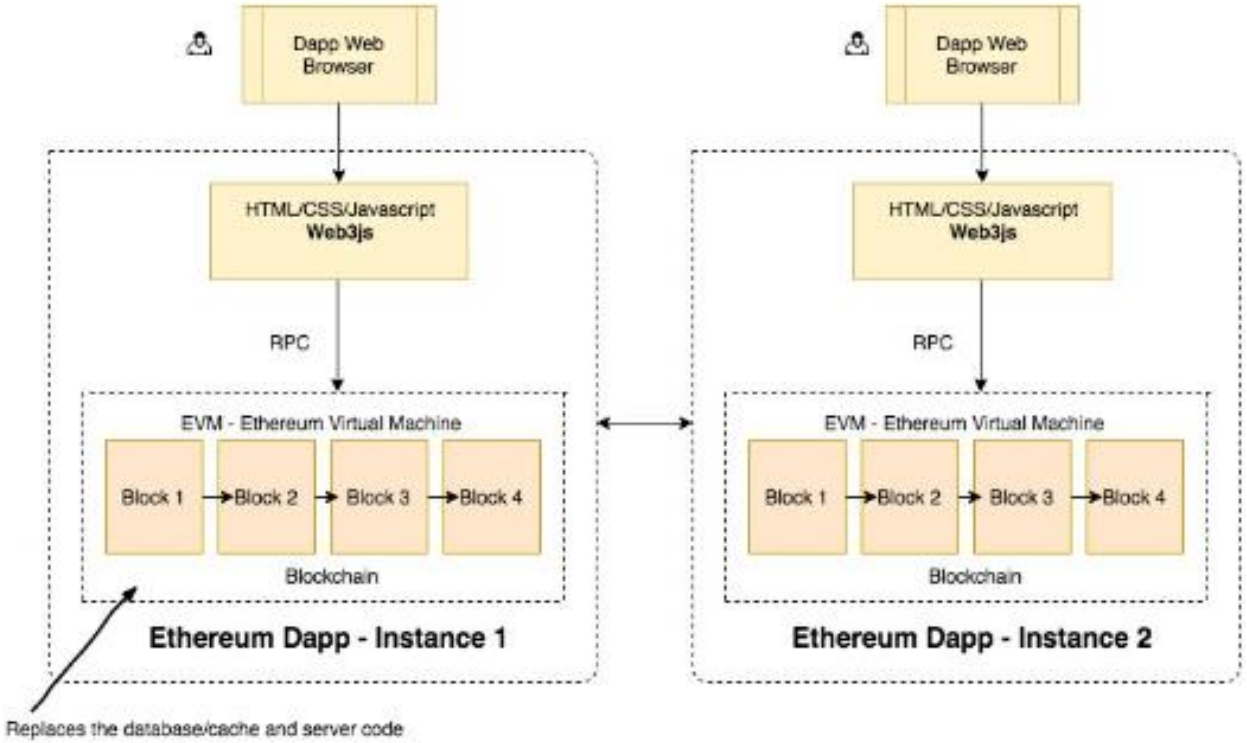


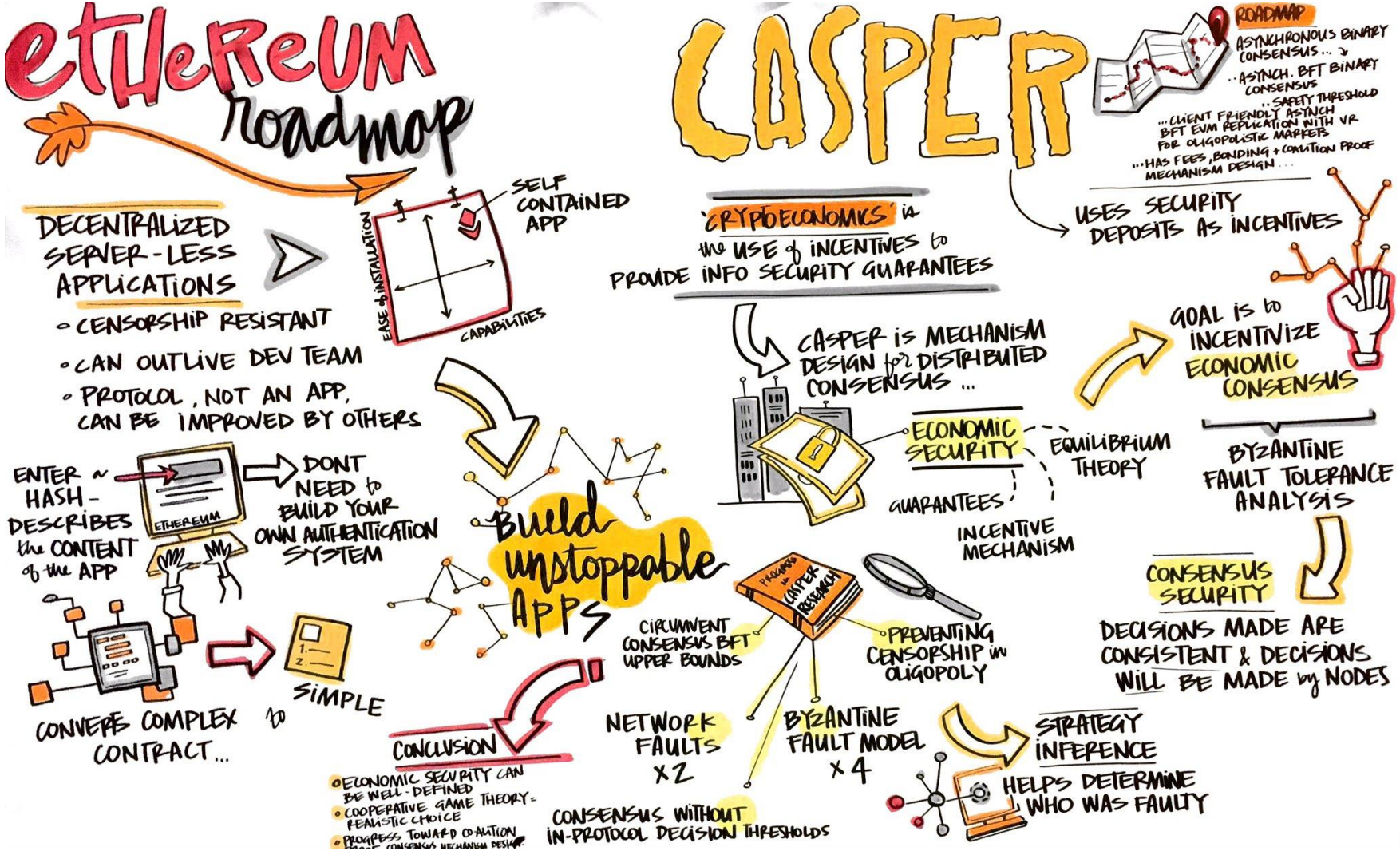
Figure 4.1: High-level DApp architecture, Source: Mahesh Murthy, medium.com

Source: Ethereum Smart Contract Development by Mayukh Mukhopadhyay

Ethereum Roadmaps

- Used to methodically improve software according to a time table
- Shows how the Ethereum Leadership is understanding the Business and Technical Environments in which Ethereum operates
- Shows how the Ethereum Leadership is addressing the challenges like growth and performance, while maintaining quality and integrity
- Informs the Ethereum Users and Developers how to anticipate the changes that will come as the Ethereum Platform continues to evolve.

Ethereum Roadmap



Ethereum Roadmaps

Frontier Release (2015)

Frontier had several main goals, all of which were met on time. Everything in this phase of Ethereum was done via the command line. Priorities at the time included the following:

- Getting mining operations running (at a reduced reward rate)
- Getting ether listed on cryptocurrency exchanges
- Establishing a live environment to test dapps
- Creating a sandbox and faucet for acquiring ether
- Allowing people to upload and execute contracts

Source: Introducing Ethereum and Solidity by Chris Dannon, Apress, 2017

Ethereum Roadmaps

Homestead Release (2016)

The Homestead release brought many more mainstream cryptocurrency enthusiasts into the fold with the Mist browser. Its characteristics are as follows:

- Ether mining goes up to 100 percent reward rate
- No network halts
- Slightly-less-beta status (fewer warnings)
- More documentation for command line and Mist

Source: Introducing Ethereum and Solidity by Chris Dannon, Apress, 2017

Ethereum Roadmaps

Metropolis (2017)

As of this writing, work is underway on Metropolis, the second phase of Ethereum protocol development. This release will be the true coming-out party for Mist, which when fully featured, will look something like a cross between Chrome and the iOS App Store. It will include several heavyweight third-party applications. By this point, Swarm and Whisper will be operational.

Source: Introducing Ethereum and Solidity by Chris Dannon, Apress, 2017

Ethereum Roadmaps

Serenity (2018)

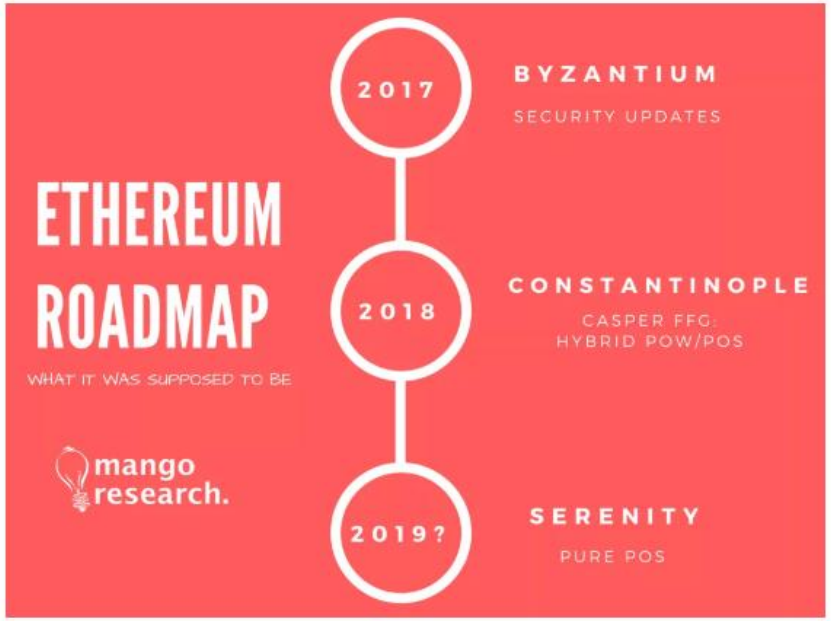
This phase is so-named for its planned transition away from proof of work and onto something less hectic: ideally, some form of proof-of-stake algorithm. For now, the tentative code name for Ethereum's POS-based consensus engine is Casper.[2] Although nobody has perfected such a consensus system yet, progress happens by the week, and mathematicians and computer scientists working in this area seem confident a breakthrough is near. Two posts that include background material on this aspect of Ethereum research can be found at the following URLs:

https://blog.ethereum.org/2015/12/24/understanding-serenity-part-i-abstraction/

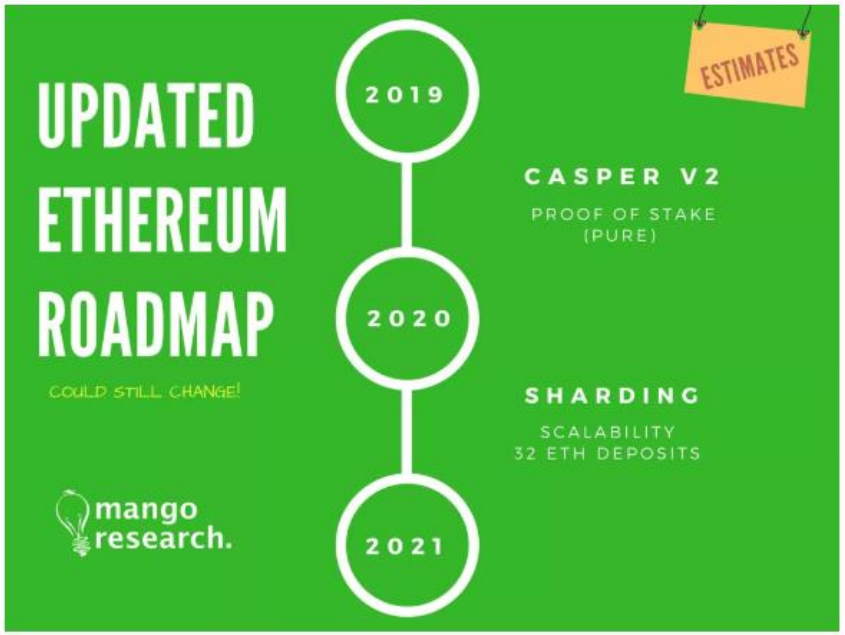
https://blog.ethereum.org/2015/12/28/understanding-serenity-part-2-casper/

Source: Introducing Ethereum and Solidity by Chris Dannon, Apress, 2017

Ethereum Roadmaps



Ethereum Roadmap Before Update



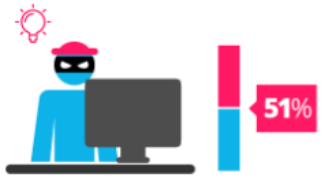
Updated Ethereum Casper Release Dates (2018 Estimates)

Ethereum Proof of Work vs. Proof of Stake

Proof of Work vs Proof of Stake

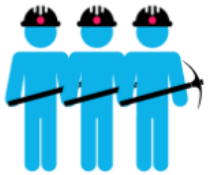
proof of work is a requirement to define an expensive computer calculation, also called mining

Proof of stake, the creator of a new block is chosen in a deterministic way, depending on its wealth, also defined as stake.



A reward is given to the first miner who solves each blocks problem.

The PoS system there is no block reward, so, the miners take the transaction fees.



Network miners compete to be the first to find a solution for the mathematical problem

Proof of Stake currencies can be several thousand times more cost effective.

Source: https://blockgeeks.com/guides/proof-of-work-vs-proof-of-stake/

Ethereum Roadmaps - Casper

Casper 2.0: The Initial Plan

The initial plan was to transition to Proof Of Stake with Casper FFG. Casper 2.0 was to be a Smart Contract that allowed you to become a validator with a deposit of 1500 ETH. The Ethereum **estimated** this release date to be somewhere in 2018.

Proof Of Stake was to be implemented first and the team would roll out Sharding after. There were separate deposit pools for Sharding and Casper.

To Summarise:

1. Casper FFG to be a Hybrid PoS and PoW chain
2. 1500 Ether deposit required to become a validator
3. Casper rolled out first, Sharding rolled out after

Source: https://www.mangoresearch.co/ethereum-roadmap-update/

Ethereum Roadmaps - Casper

Casper 2.0: The Initial Plan

The initial plan was to transition to Proof Of Stake with Casper FFG. Casper 2.0 was to be a Smart Contract that allowed you to become a validator with a deposit of 1500 ETH. The Ethereum **estimated** this release date to be somewhere in 2018.

Proof Of Stake was to be implemented first and the team would roll out Sharding after. There were separate deposit pools for Sharding and Casper.

To Summarise:

1. Casper FFG to be a Hybrid PoS and PoW chain
2. 1500 Ether deposit required to become a validator
3. Casper rolled out first, Sharding rolled out after

Source: https://www.mangoresearch.co/ethereum-roadmap-update/

Ethereum – Proof of Stake

Casper 2.1: *The Confusion over the Releases*

Due to some misleading posts and misunderstood comments, several people are confused. These are the two primary impressions that people have in regard to the Casper update:

1. Casper and Sharding will be combined and launched together.
2. Sharding will now be prioritized over Proof Of Stake

This is not true at all. And it's important that expectations are set right.

Casper 2.1: *The Real Roadmap*

The plan for Casper FFG requiring 1500 ETH deposits will be scrapped. Casper V2 will be implementing a "beacon chain" – onto which Casper and Sharding will be merged (here is where people get confused).

This does not mean that Casper and Sharding will be launched on the beacon chain together. It simply means that Casper and Sharding will be implemented on the same chain. So, Casper could come first, and Sharding be implemented much later. Or vice-versa.

Source: https://www.mangoresearch.co/ethereum-roadmap-update/

Ethereum – Proof of Stake - Inside Ethereum's Plan To Reduce Energy Consumption by 99%

One of the most interesting things with respect to PoS is the fact that given validators are not expending as much energy(compared to PoW) to secure the network, the reward may be significantly lower. According to the Casper Github wiki:

> Because of the lack of high electricity consumption, there is **not as much need to issue as many new coins** in order to motivate participants to keep participating in the network.

With Proof-Of-Work, miners race to process the same set of transactions. However, Proof-Of-Stake randomly picks validators to process and secure transactions.

Source: https://www.ccn.com/inside-ethereums-plan-to-reduce-energy-consumption-by-99/

Ethereum Roadmaps – Transitioninng from PoW to PoS

Transitioning from a proof-of-work to a proof-of-stake consensus algorithm. As a consensus system , proof of work is effective but expensive from a power-consumption perspective. Securing consensus without mining would reduce electricity waste as well as the need for the inflationary issuance scheme.

Faster block times should result from proof of stake, resulting in greater granularity of data and efficiency without a loss of security or risk of centralization.

Economic finality. As covered in Chapter 3, the promise of Ethereum for enterprises is a decentralized system for transaction settlement finality. Proof-of-stake systems might include roles for validator nodes that *fully commit* to a block, meaning they lose their ETH balance (which could be millions of dollars) if they collude to propagate a false block.

Source: Introducing Ethereum and Solidity by Chris Dannon, Apress, 2017

Ethereum Roadmaps – Transitioning from PoW to PoS

Scalability is a problem when full nodes require the computing resources they do today. The large blockchain, 1 GB DAG, and intensive CPU or GPU requirements make smartphones and other low-power devices a no-go for Ethereum node daemons. To read the team's white paper on scalability, visit https://github.com/vbuterin/scalability_paper/blob/master/scalability.pdf.

Another vital read about scalability is the use of so-called chain fibers, at www.reddit.com/r/ethereum/comments/31jm6e/new_ethereum_blog_post_by_dr_gavin_wood/.

Source: Introducing Ethereum and Solidity by Chris Dannon, Apress, 2017

Ethereum Roadmaps – Transitioning from PoW to PoS

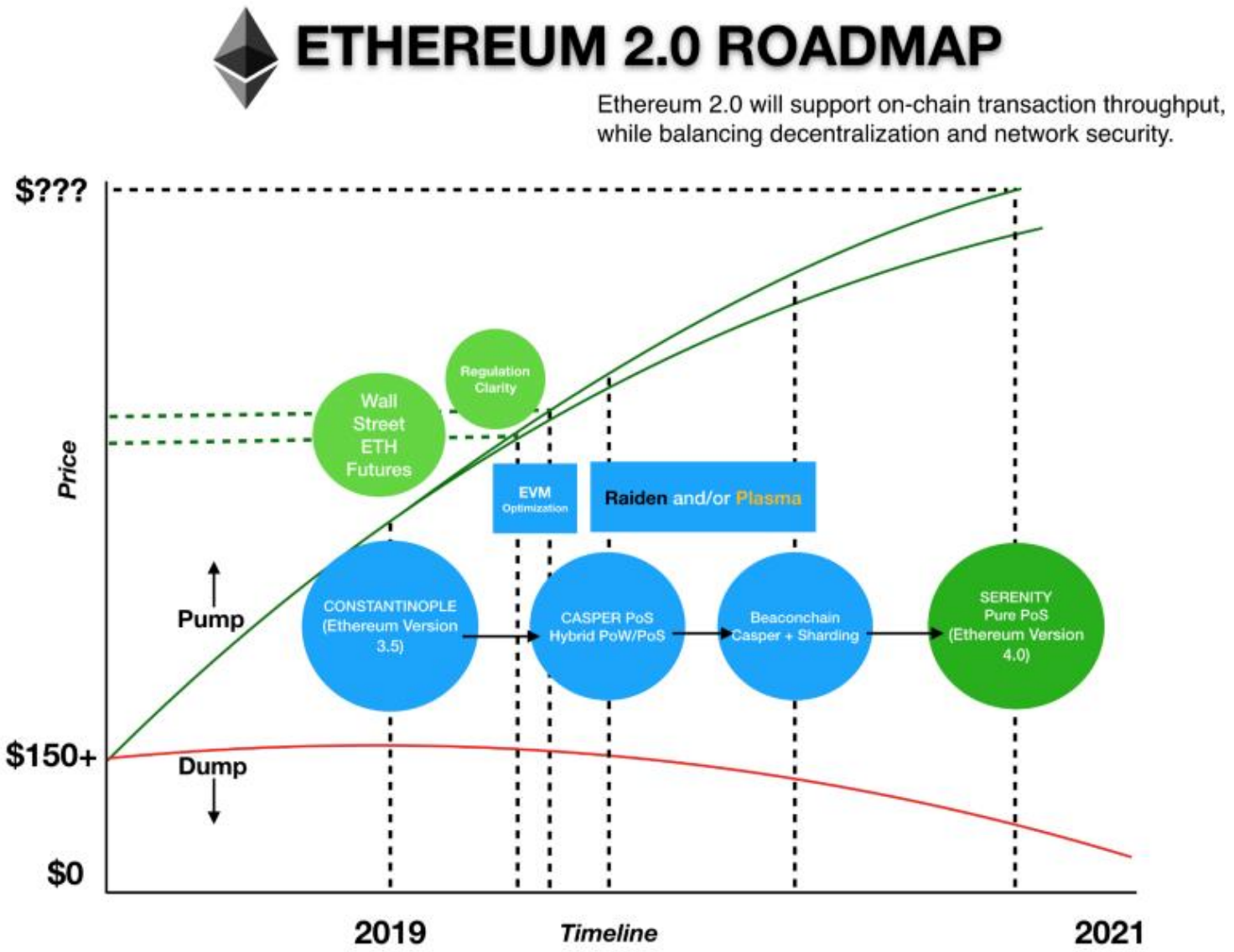
Sharding blockchain data and enabling cross-shard communication is another crucial element of scaling. *Sharding* is the process of breaking up a single chunk of data across databases, in such a way that it can be reassembled when needed. Blockchains don't shard. However, it should be feasible to let different parts of the EVM state be stored by different nodes, and to build applications that can address them there.

Being resistant to censorship in the form of attempts by validator nodes, in a proof-of-work scheme, to collude across all shards in order to block certain transactions from reaching finality. This already exists in Ethereum 1.0, but will be strengthened in subsequent releases.

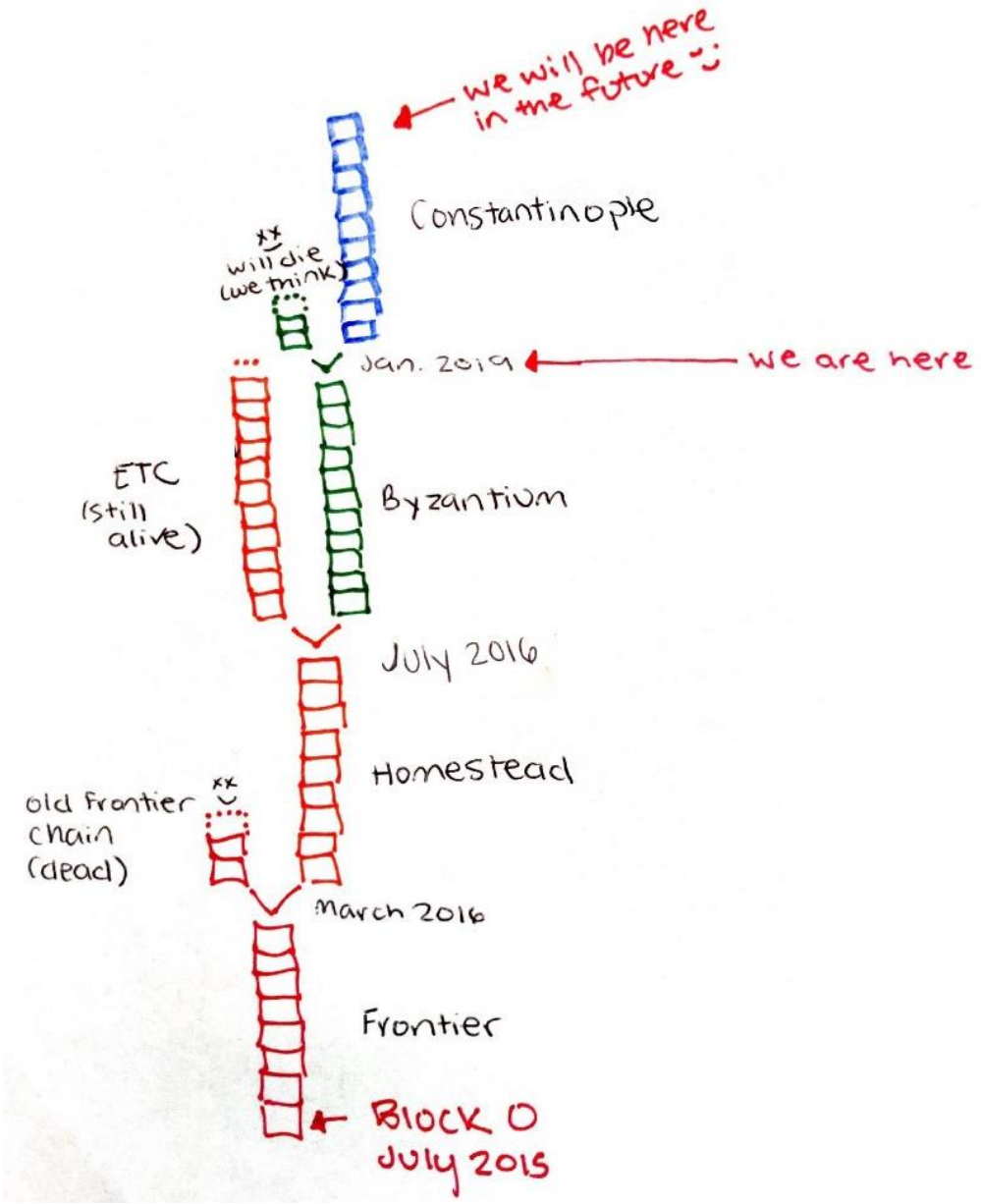
The Mauve Paper is located at http://vitalik.ca/files/mauve_paper.html .

Source: Introducing Ethereum and Solidity by Chris Dannon, Apress, 2017

Ethereum Roadmaps

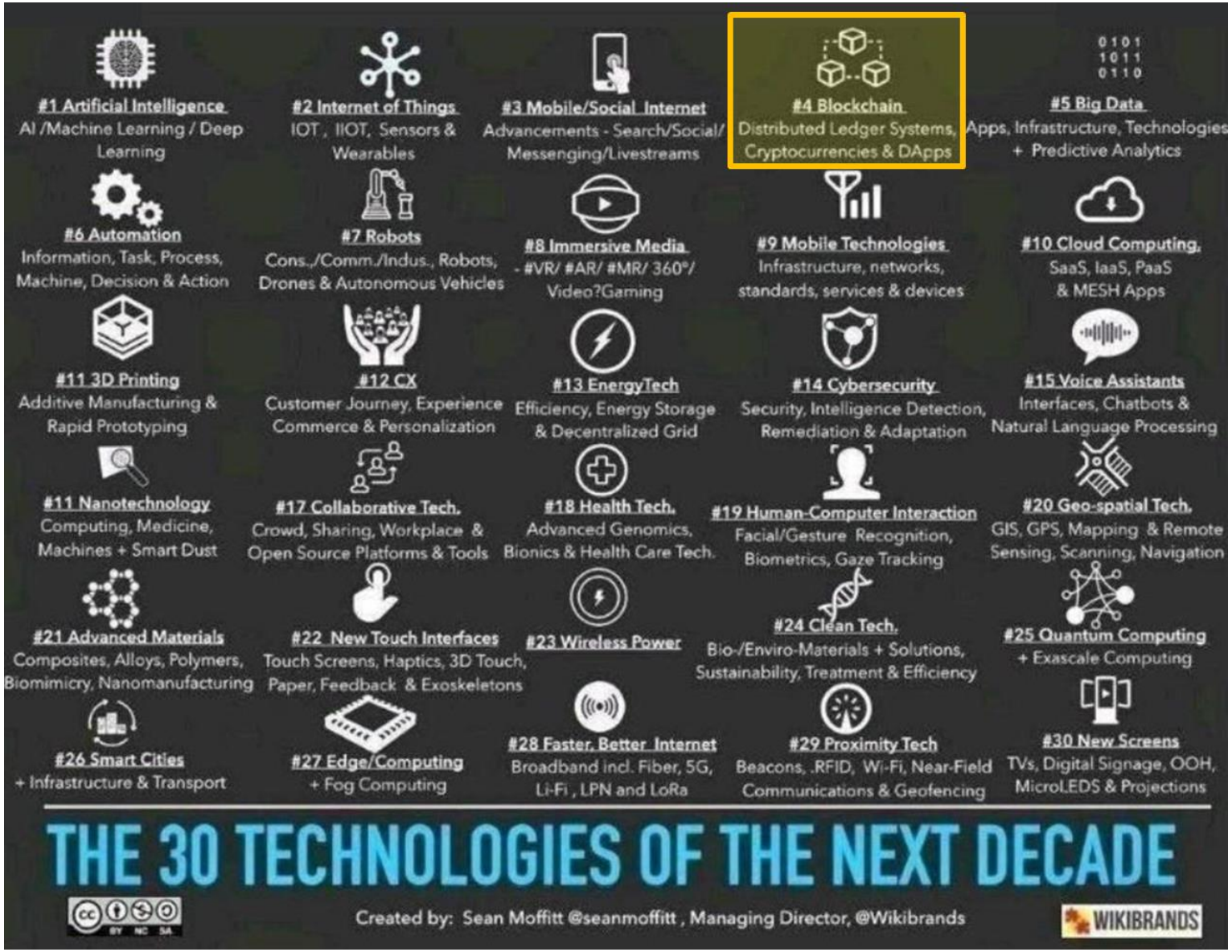


Ethereum Roadmaps

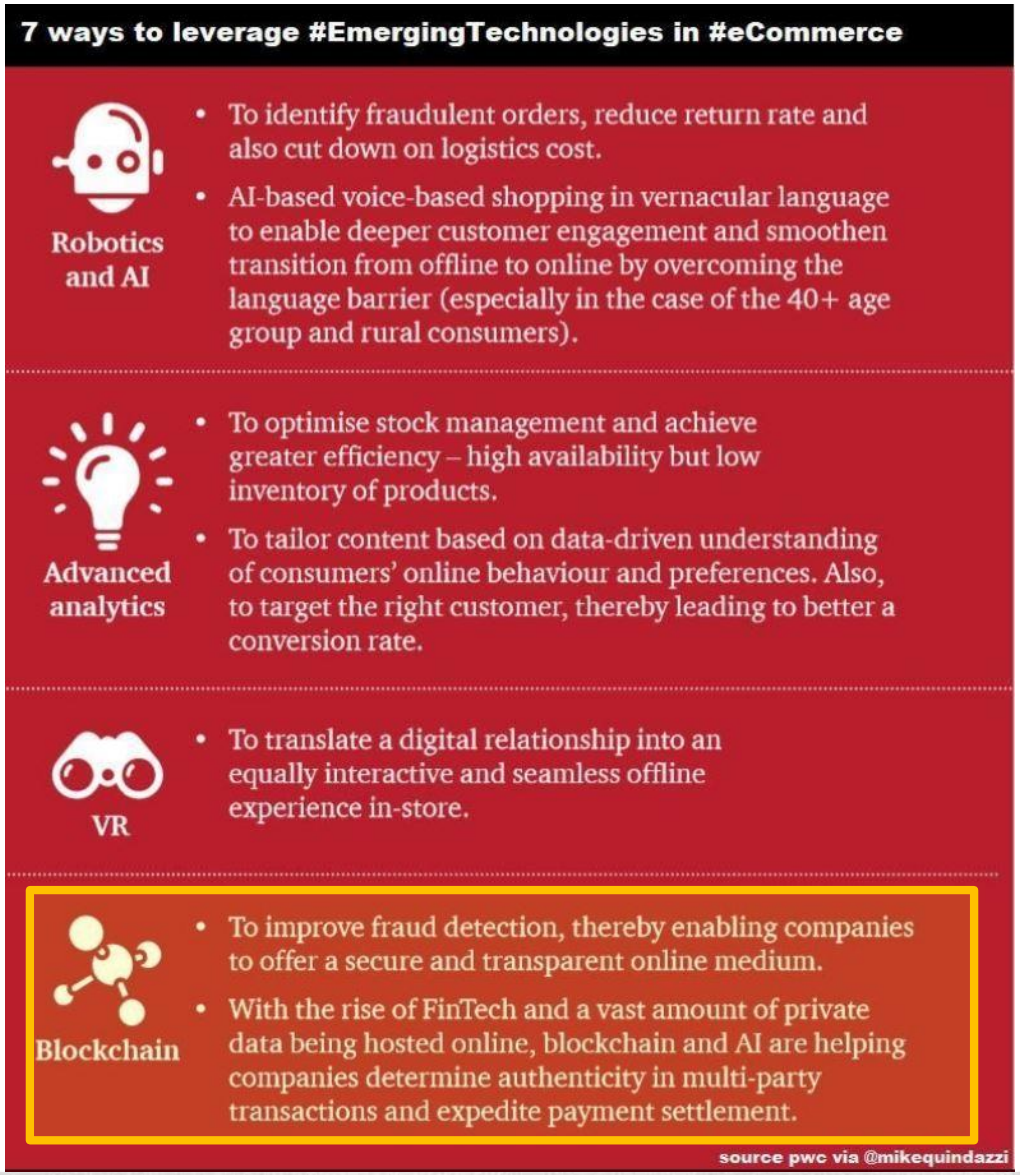


Topic 6: Blockchain Beyond Bitcoin

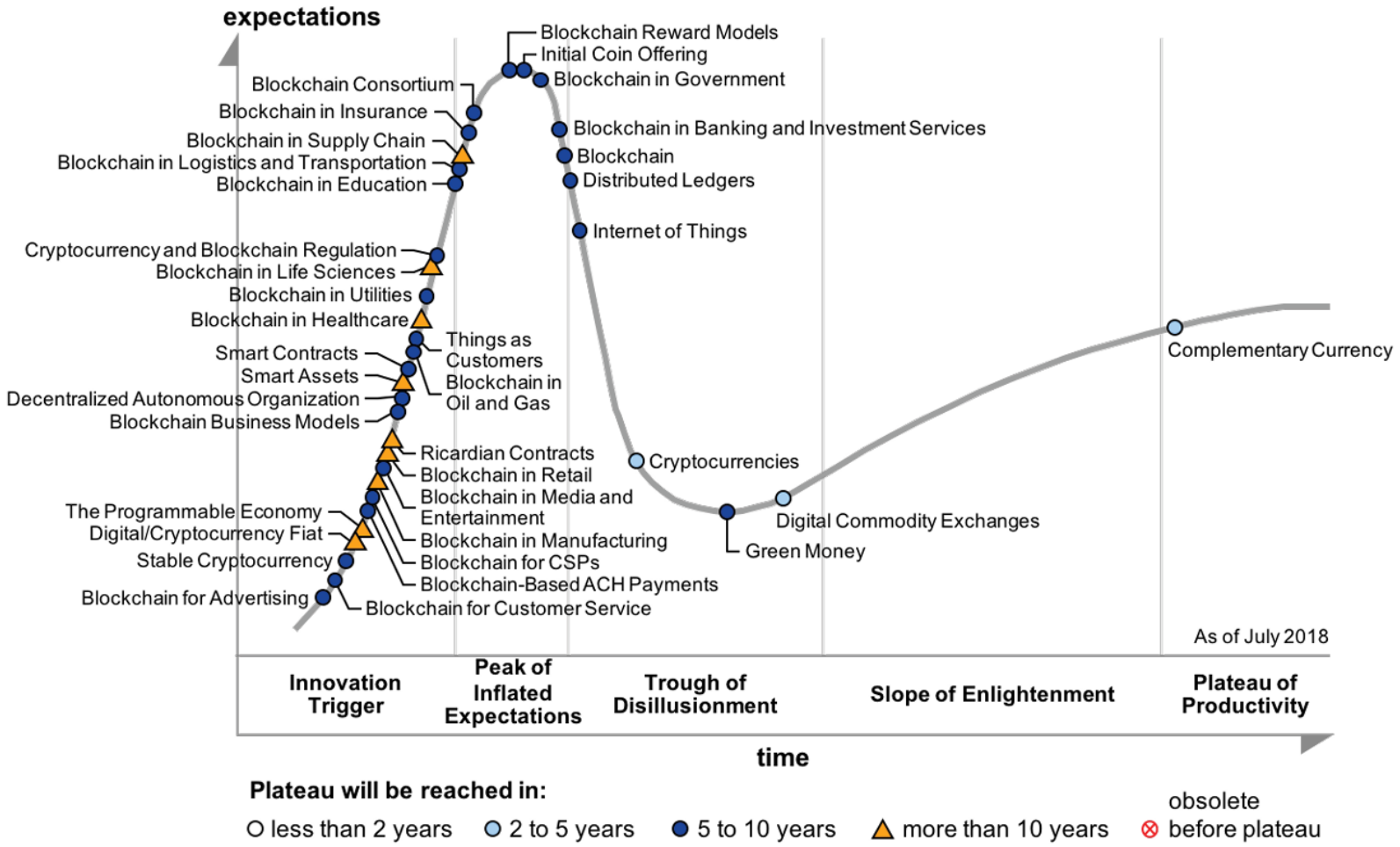
Blockchain as an Emerging Technology



Blockchain as an Emerging Technology

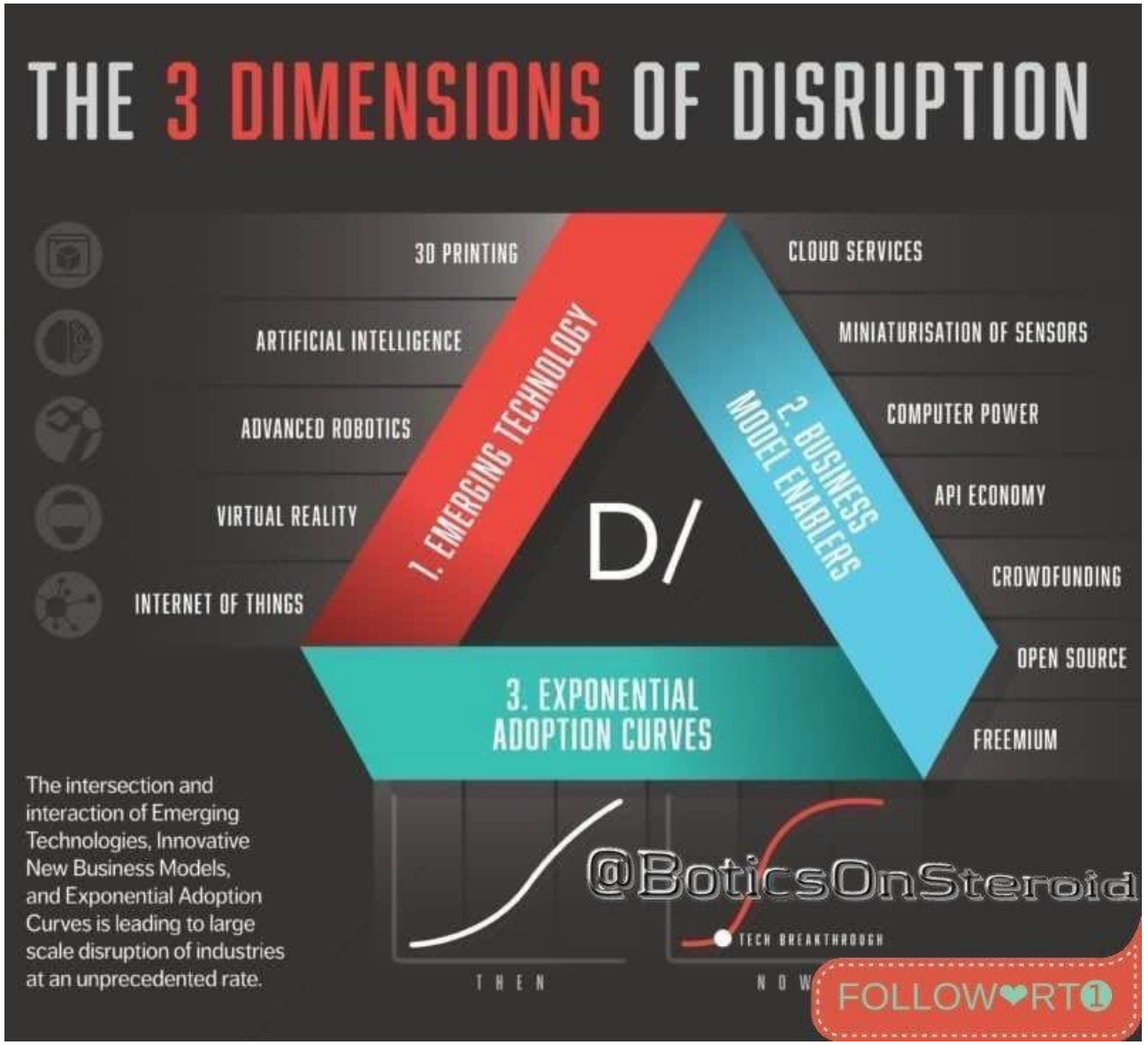


Blockchain on the Gartner Hype Cycle Curve - 2018

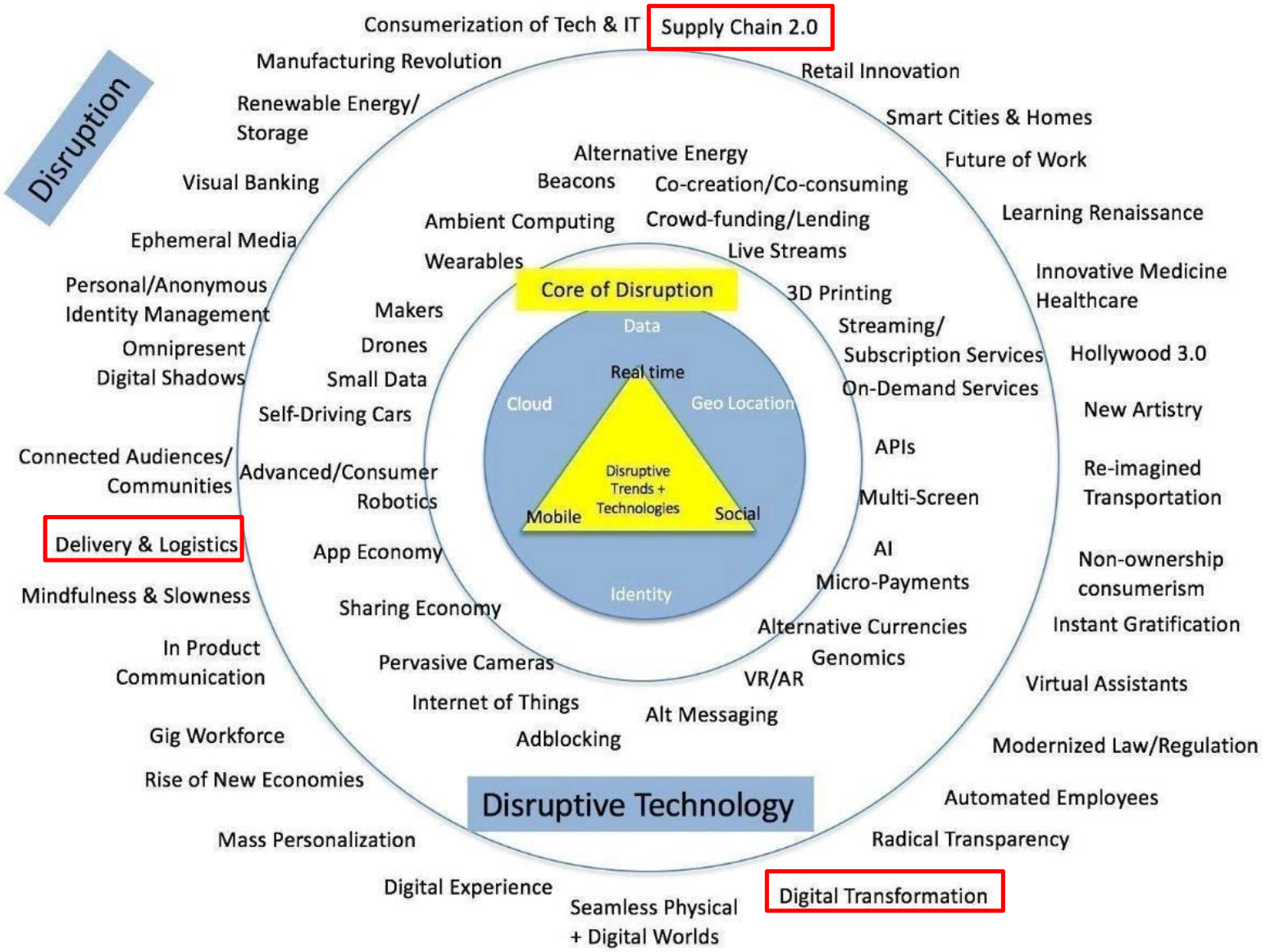


gartner.com/SmarterWithGartner

Disruption



Disruption and Disruptive Technologies



Smart Contracts and Supply Chain Management

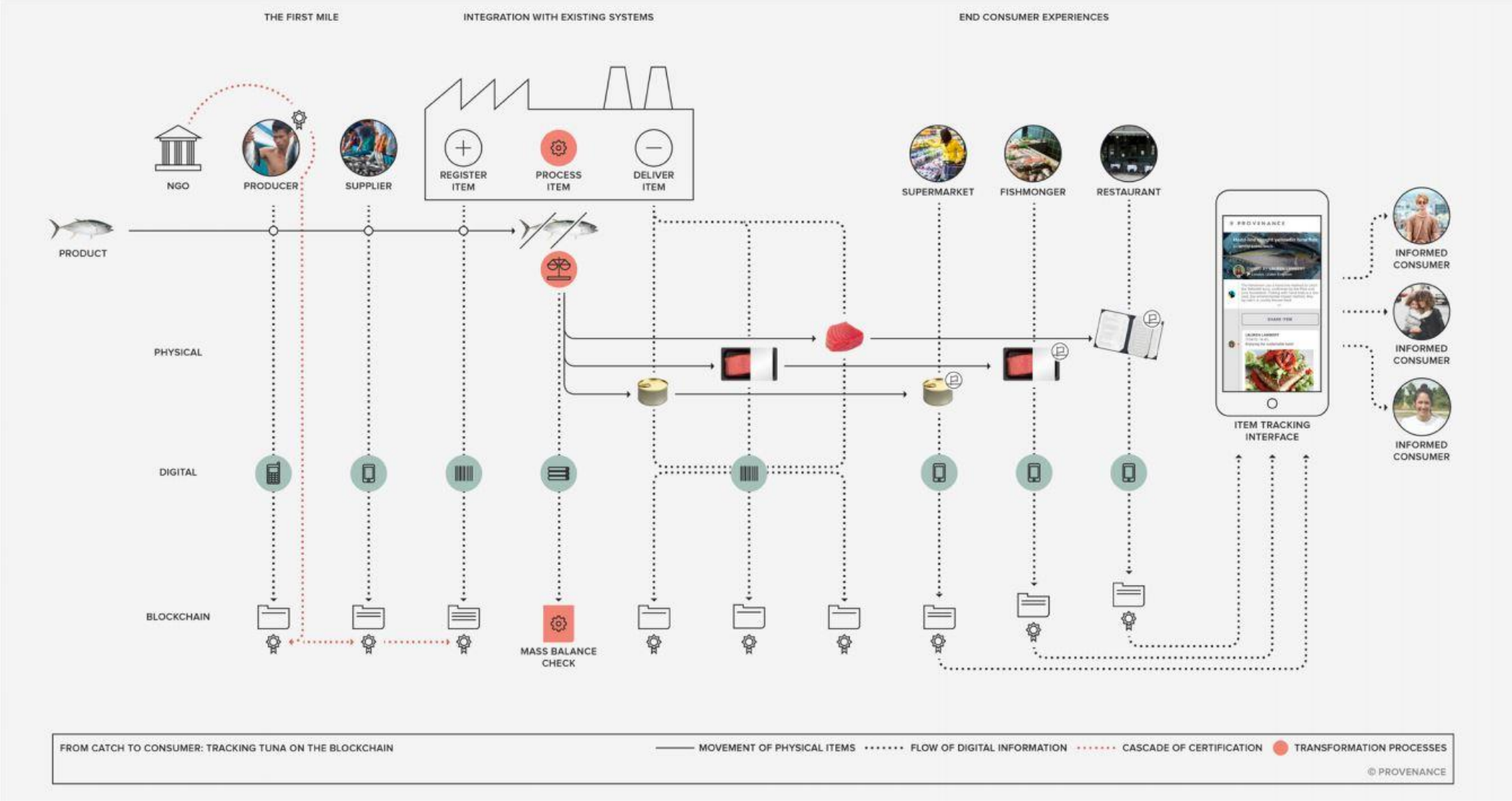


Figure 1: This chart shows how Provenance uses blockchain technology to not only permanently record certifications of supply chain data for tuna (up through sale), but also those of the participating NGOs tasked with ensuring the catch is slavery-free. (Source: Provenance)

Source:Provenance.

Lunch

Categories of Blockchain Uses and Solutions

6 Distinct Categories of Blockchain Use Cases

There are six distinct categories of blockchain use cases addressing two major needs.

Record keeping: storage of static information

Transactions: registry of tradeable information

1 Static registry

- Distributed database for storing reference data

Example
- Land title
- Food safety and origin
- Patent

2 Identity

- Distributed database with identity-related information
- Particular case of static registry treated as a separate group of use cases due to extensive set of identity-specific use cases

Example
- Identity fraud
- Civil-registry and identity records
- Voting

3 Smart contracts

- Set of conditions recorded on a blockchain triggering automated, self-executing actions when these predefined conditions are met

Example
- Insurance-claim payout
- Cash-equity trading
- New-music release

4 Dynamic registry

- Dynamic distributed database that updates as assets are exchanged on the digital platform

Example
- Fractional investing
- Drug supply chain

5 Payments infrastructure

- Dynamic distributed database that updates as cash or cryptocurrency payments are made among participants

Example
- Cross-border peer-to-peer payment
- Insurance claim

6 Other

- Use case composed of several of the previous groups
- Standalone use case not fitting any of the previous categories

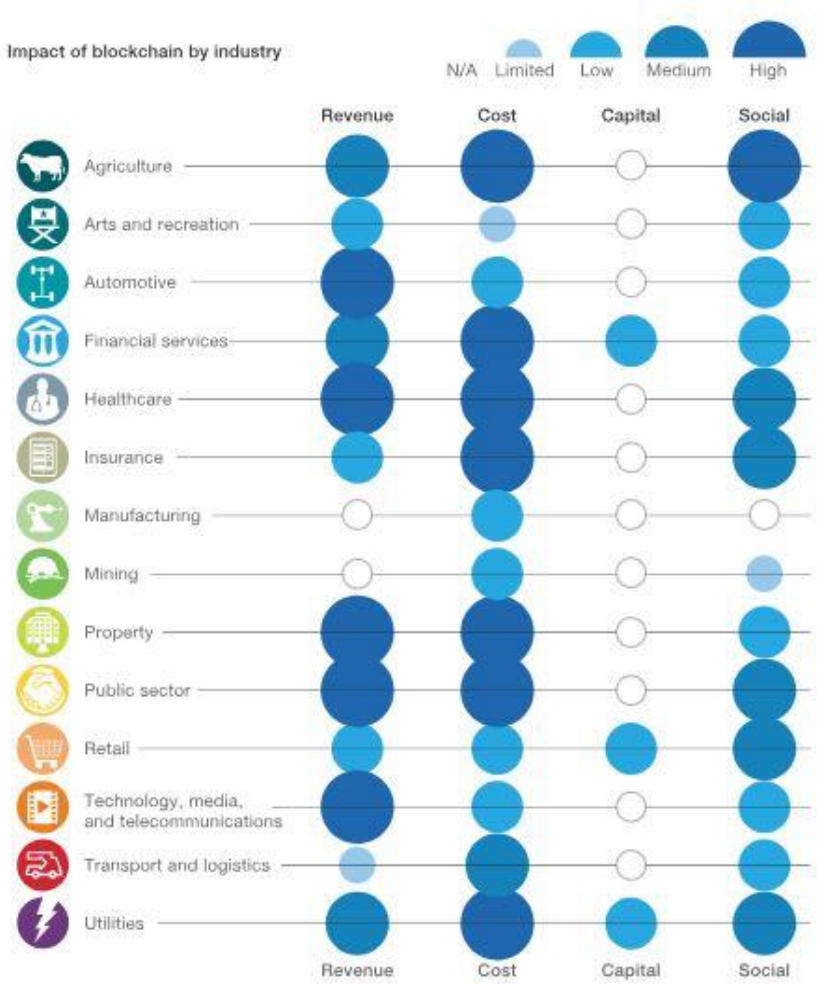
Example
- Initial coin offering
- Blockchain as a service

McKinsey&Company

Case Study: Blockchain Use Cases Across Industries

Exhibit 4

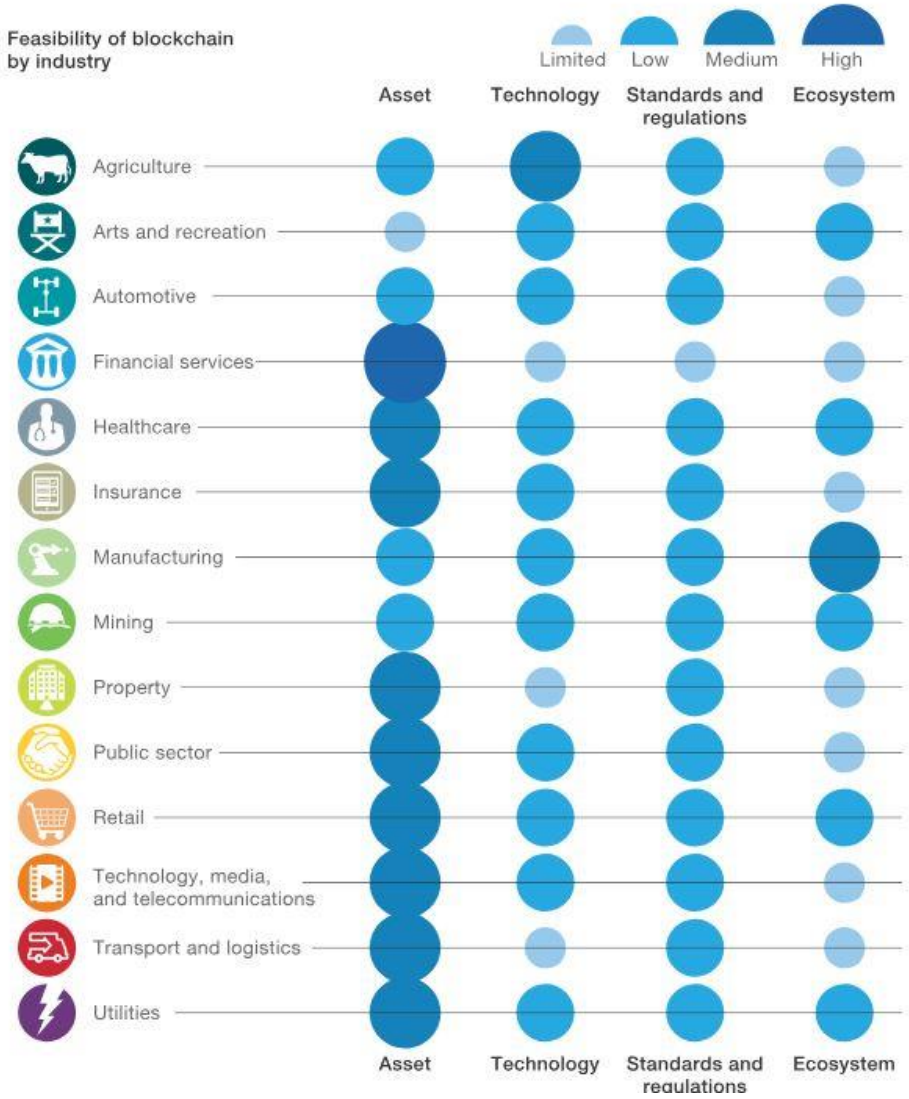
The value at stake from blockchain varies across industries.



McKinsey&Company

Source: https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/blockchain-beyond-the-hype-what-is-the-strategic-business-value

Case Study: Blockchain Feasibility Across Industries



Source: https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/blockchain-beyond-the-hype-what-is-the-strategic-business-value

Why Is Blockchain an Interesting and Important Technology?

Why Blockchain?

Why Is Blockchain Important?

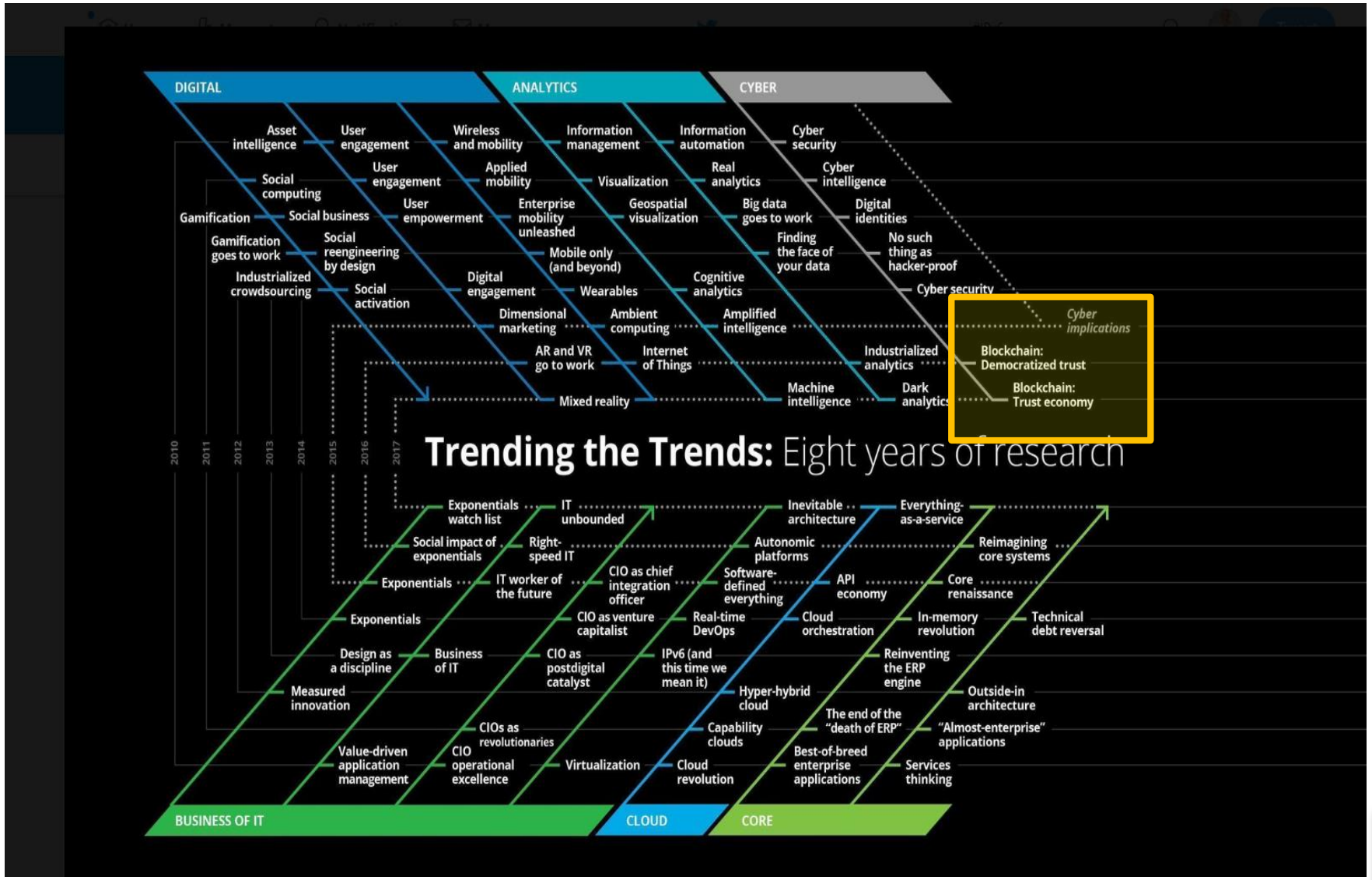
- Accessible
- Open source
- Easily provides three challenging elements of the **Parkerian Hexad** model for security:
  - **Authenticity**
  - **Control**
  - **Utility**
- Immutable transactions
- Decentralized
- It WORKS!
- Business enabler
- Reduces risk of computer fraud
- It is being widely adopted for trusted computing
- Blockchain developers and architects are in great demand: for every Blockchain professional there are 14 open positions

Donn B. Parker

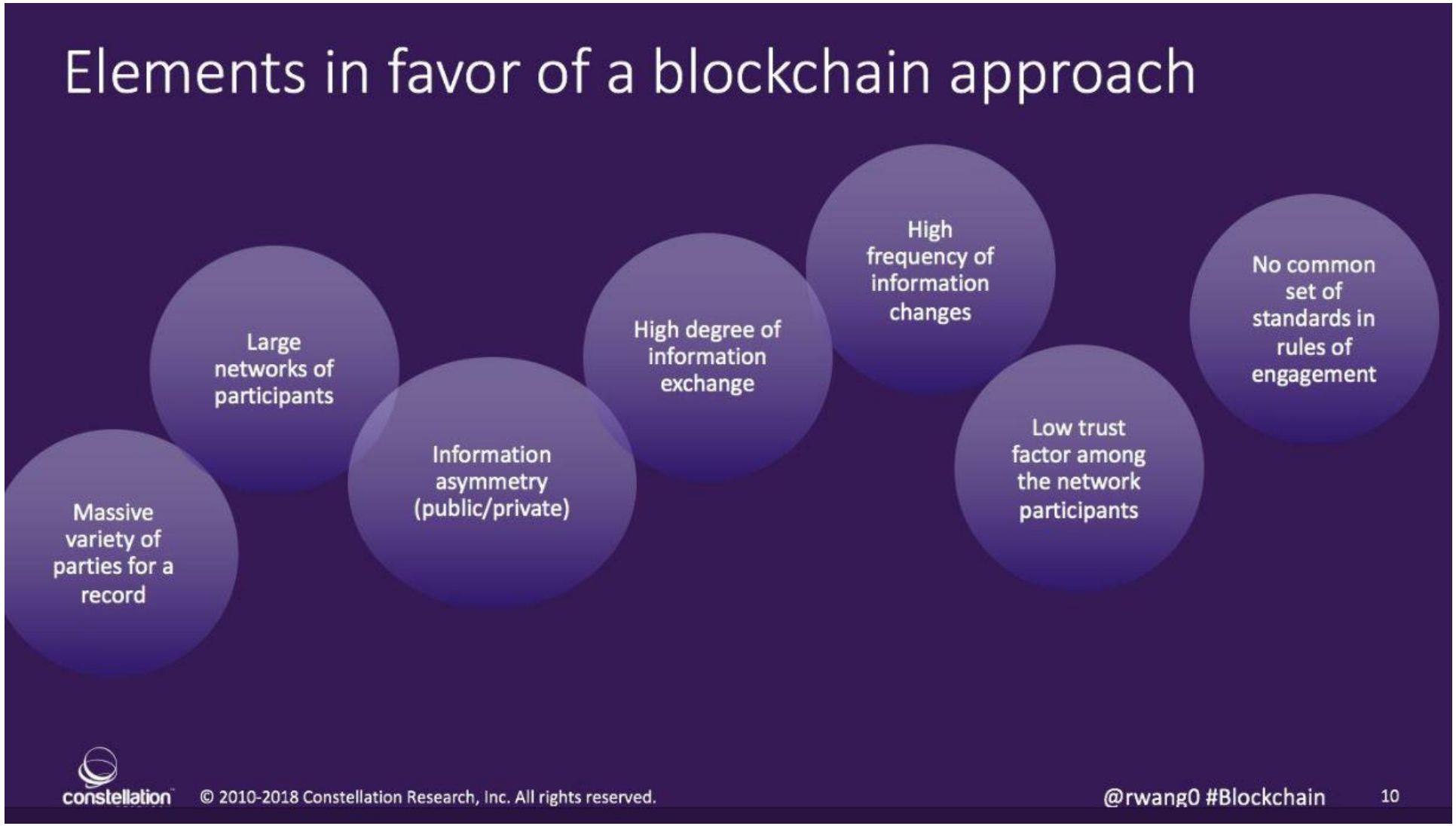
Parkerian Hexad

Donn B. Parker

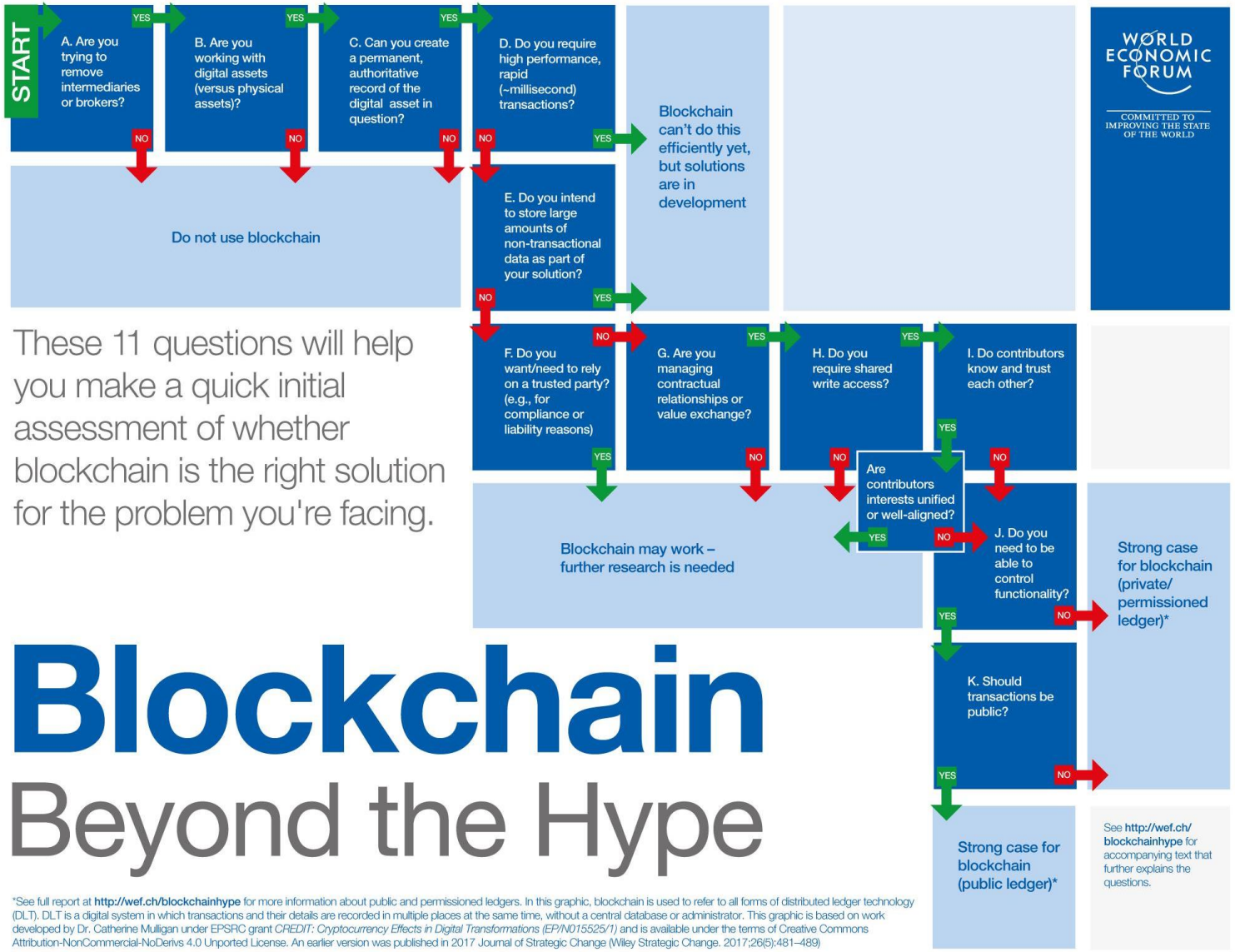
Trending the Trends – 8 Years of Research



Elements that Favor a Blockchain Approach



Blockchain beyond the Hype



WORLD ECONOMIC FORUM
COMMITTED TO IMPROVING THE STATE OF THE WORLD

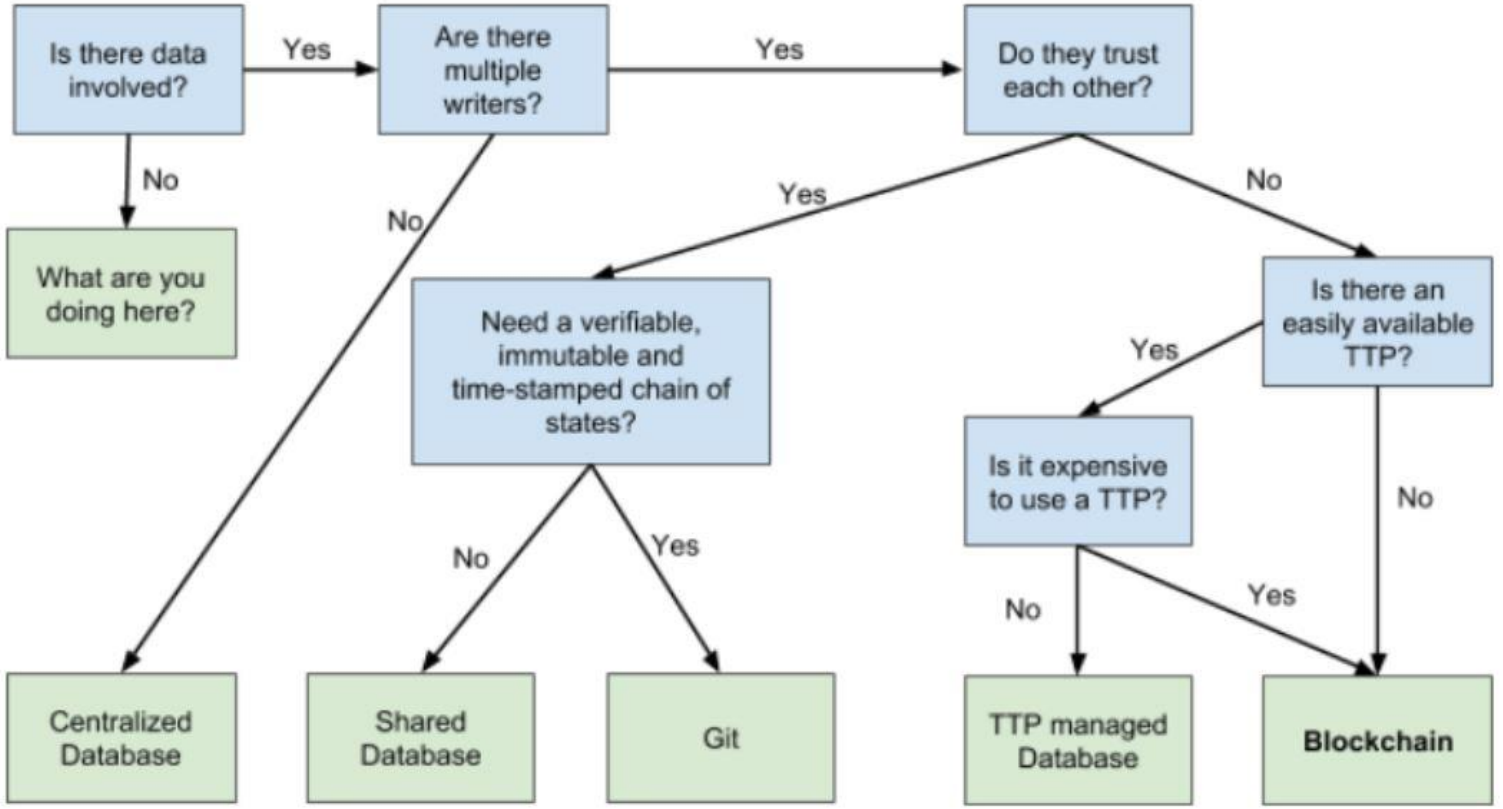
These 11 questions will help you make a quick initial assessment of whether blockchain is the right solution for the problem you're facing.

Blockchain
Beyond the Hype

*See full report at **http://wef.ch/blockchainhype** for more information about public and permissioned ledgers. In this graphic, blockchain is used to refer to all forms of distributed ledger technology (DLT). DLT is a digital system in which transactions and their details are recorded in multiple places at the same time, without a central database or administrator. This graphic is based on work developed by Dr. Catherine Mulligan under EPSRC grant *CREDIT: Cryptocurrency Effects in Digital Transformations (EP/N015525/1)* and is available under the terms of Creative Commons Attribution-NonCommercial-NoDerivs 4.0 Unported License. An earlier version was published in 2017 Journal of Strategic Change (Wiley Strategic Change. 2017;26(5):481–489)

To Blockchain or Not to Blockchain

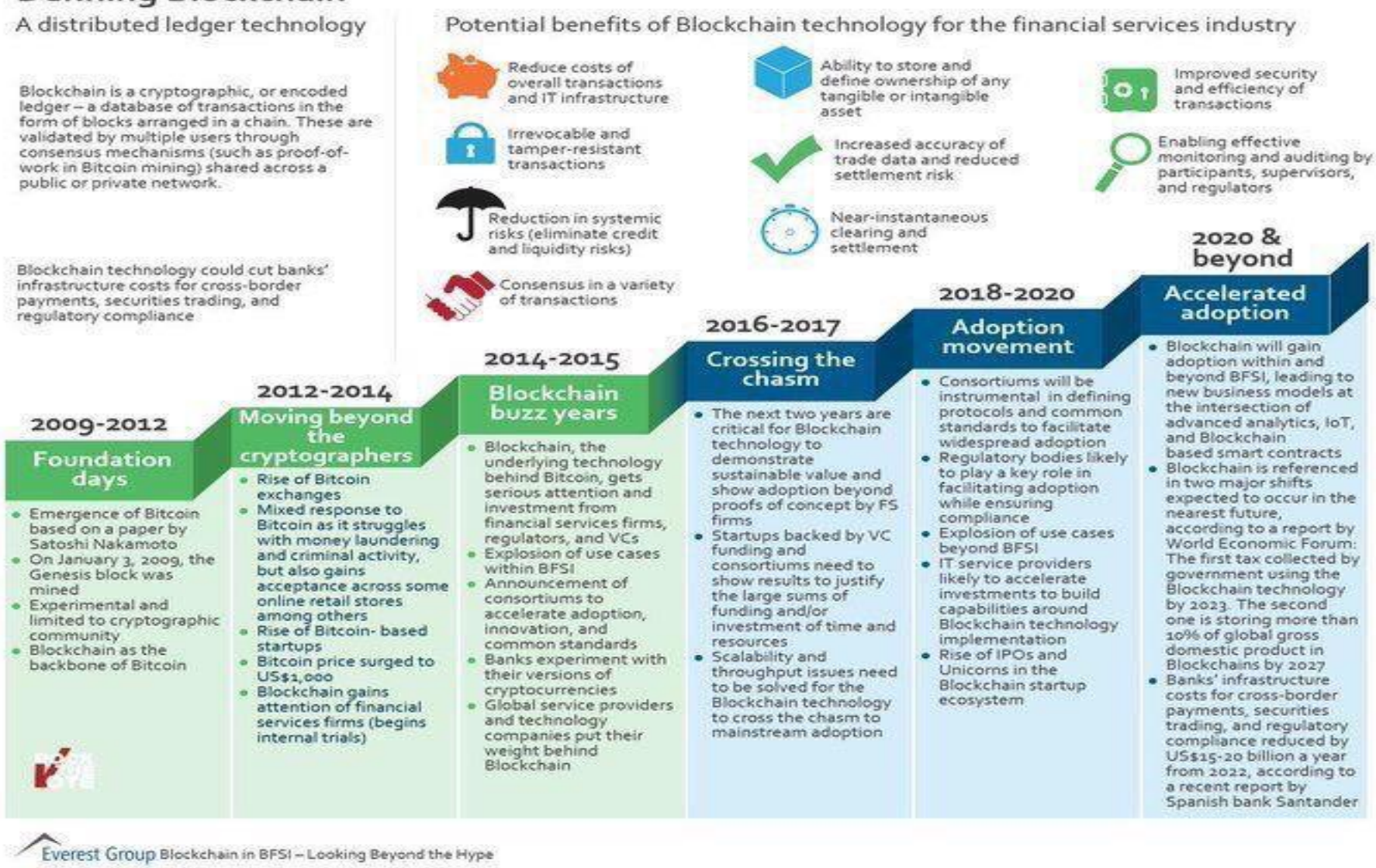
If you are a little lost, don't worry, here is a visual framework that will help you assess whether a Blockchain is something you should be looking into:



Source: To Blockchain or not to Blockchain? https://medium.com/causys/to-blockchain-or-not-to-blockchain-aed05bf08150 Hats off to the author, Thomas Ferry of Causys

Use Cases

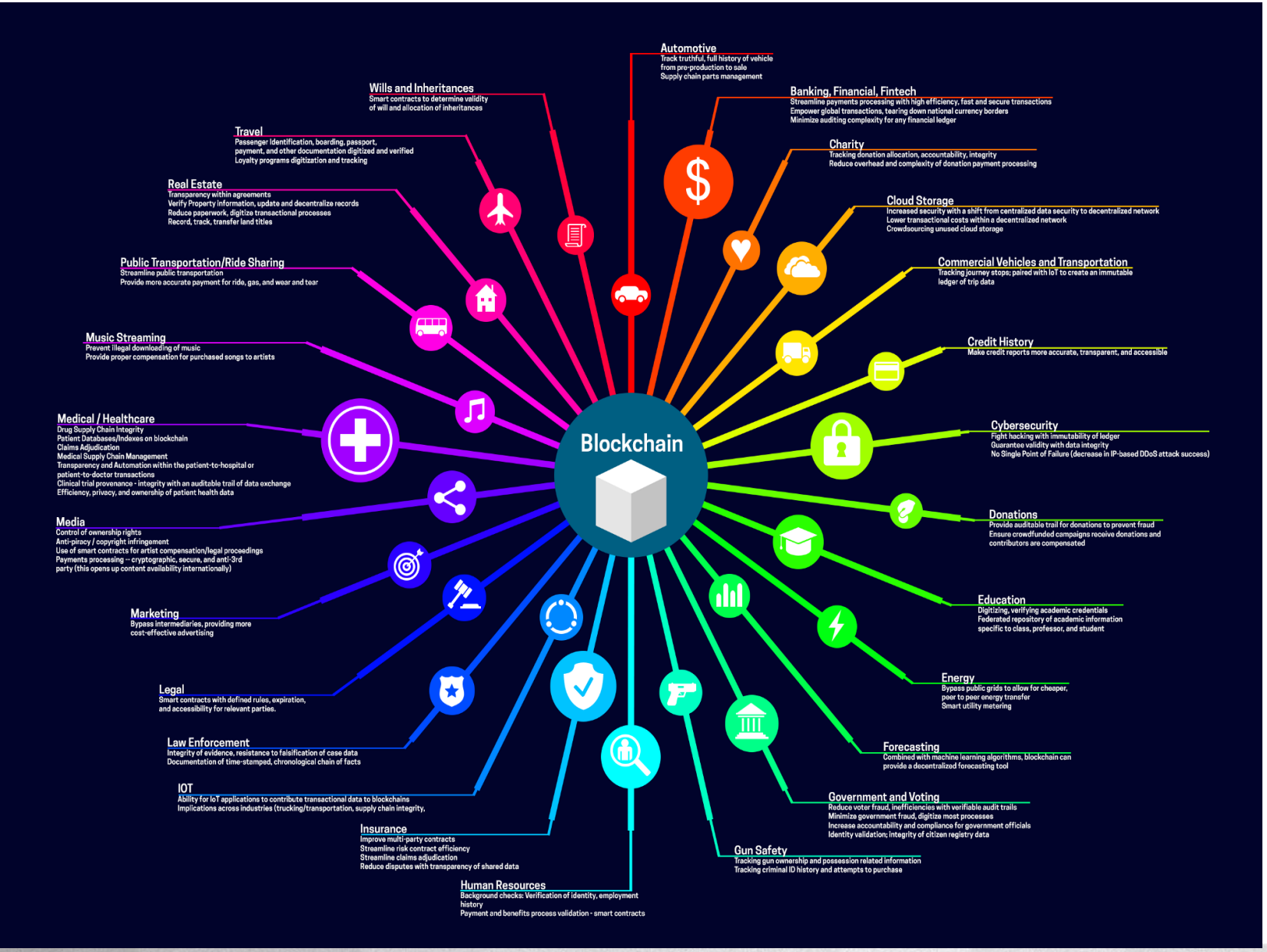
Blockchain Use Evolution



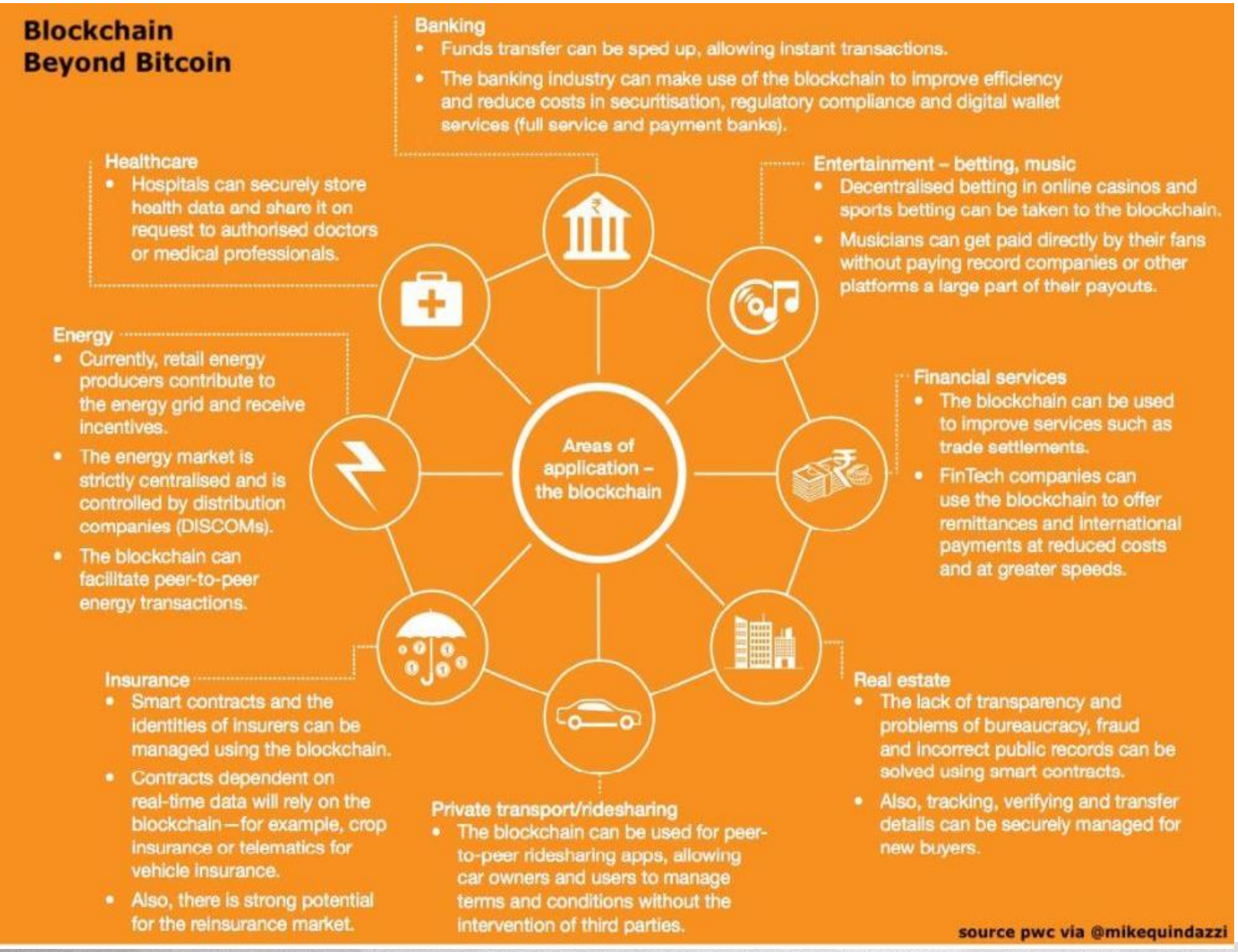
Blockchain Use Cases – Comprehensive Analysis and Start-ups Involved

2019 NORTH AMERICA CACS™ AN ISACA EVENT

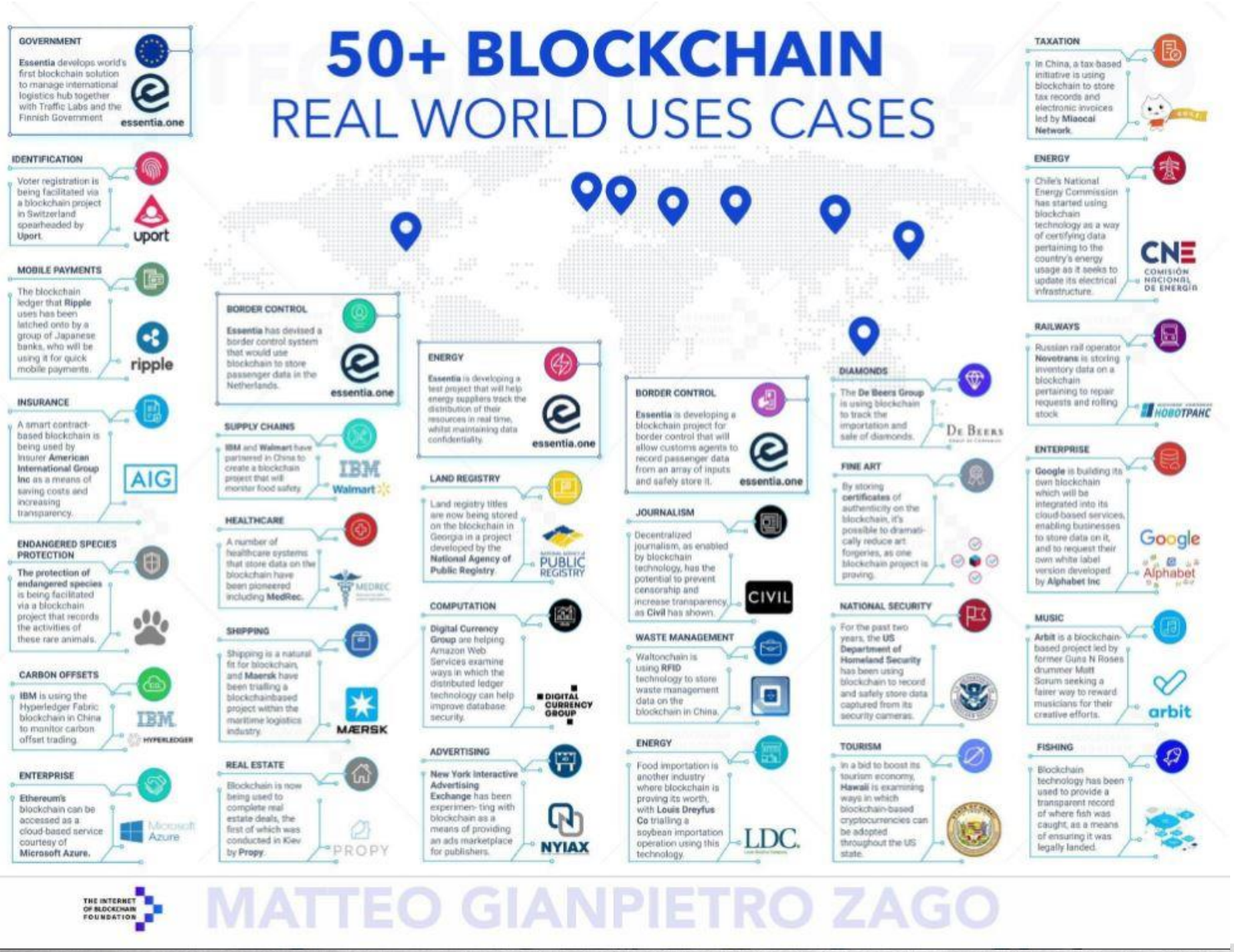
Blockchain Uses Cases by Industry



Blockchain Beyond Bitcoin



50+ Blockchain Real-World Use Cases



THE INTERNET OF BLOCKCHAIN FOUNDATION

MATTEO GIANPIETRO ZAGO

More Blockchain Use Cases

Non-Financial Use Cases

Digital Content/Documents, Storage & Delivery

BitProof, Blockcai, Ascribe, ArtPlus, Chainy.Link, Stampery, Blocktech (Alexandria), Bisantyum, Blockparti, The Rudimental, BlockCDN

Authentication & Authorization

The Real McCoy, Degree of Trust, Everpass, BlockVerify,

Digital Identity

Sho Card, Uniquid, Onename, Trustatom

Marketplace

Providing premium rights & brand based coins: MyPowers

Smart Contracts

Otonomos, Mirror, Symbiont, New system Technologies

Real Estate

Factom

Diamonds

Everledger

Gold & Silver

BitShares, Real Asset Co., DigitalTangible (Serica), Bit Reserve

Reviews/Endorsement

TRST.im, Asimov (recruitment services), The World Table

Blockchain in IoT

Filament, Chimera-inc.io, ken Code – ePlug

App Development

Proof of ownership for modules in app development: Assembly

Network Infrastructure & APIs

Ethereum, Eris, Codius, NXT, Namecoin, Colored Coins, Hello Block, Counterparty, Mastercoin, Corona, Chromaway, BlockCypher

Other

Prediction platform: Augur
Election Voting: Follow My Vote
Patient Records management: BitHealth

Financial Use Cases

Currency Exchange & Remittance

Coinbase (Wallet), BitPesa, Billion, Ripple, Stellar, Kraken, Fundrs.org, MeXBT, CryptoSigma

P2P Transfers

BTC Jam, Codius, BitBond, BitnPlay (Donation), DeBuNe (SME's B2B transactions)

Ride Sharing

La'zooz

Data Storage

Storj.io, Peernova

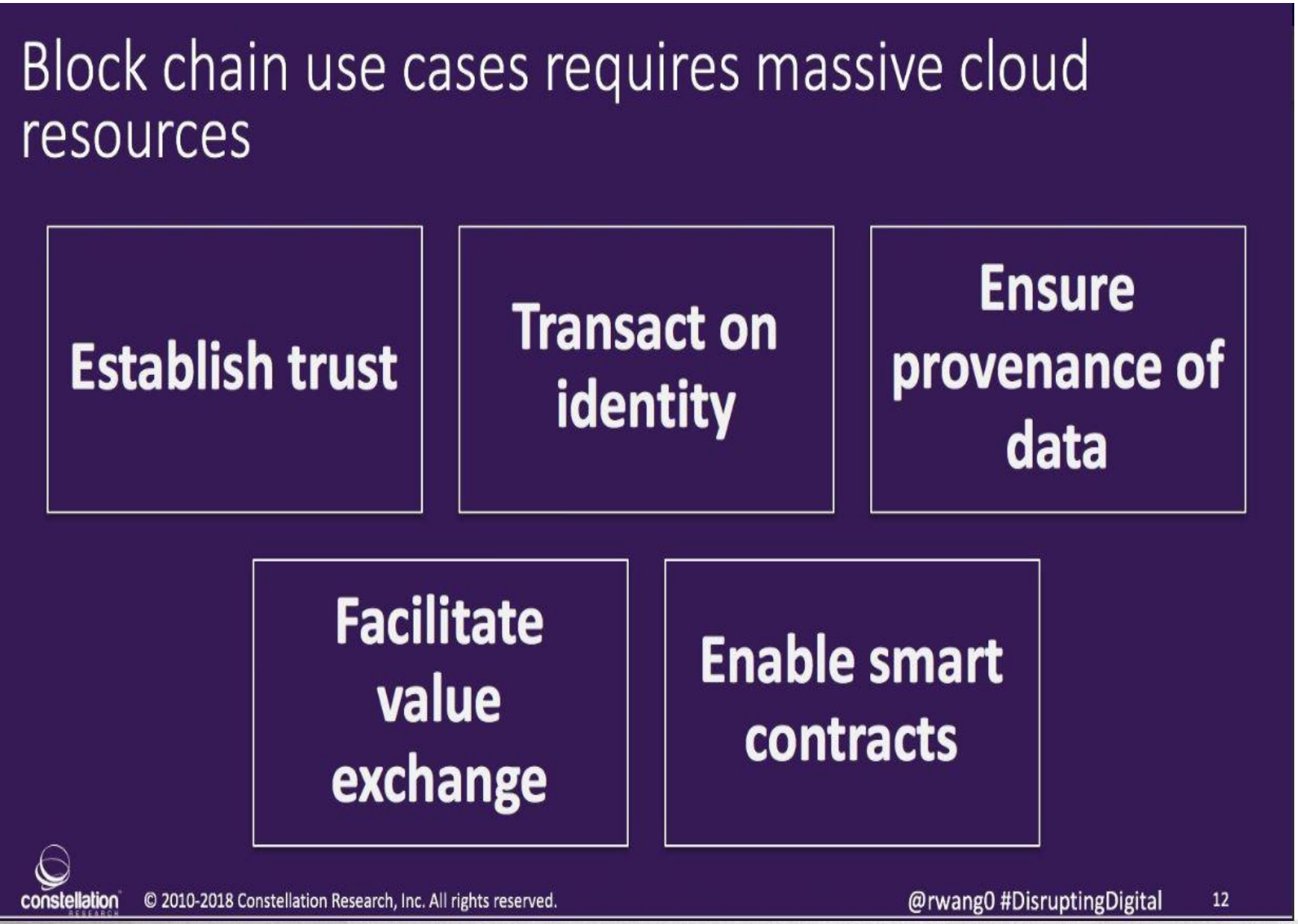
Trading Platforms

equityBits, Spritzle, Secure Assets, Coins-e, DXMarkets, MUNA, Kraken, BitShares

Gaming

PlayCoin, Play(on DACx platform), Deckbound

Blockchain Use Case Considerations



Blockchain Solution Examples

Definition of a Smart Contract - 1996

The smart contract was formally defined by Nick Szabo in his 1996 paper titled, *Smart Contracts: Building Blocks for Digital Markets*. He described it as follows:

"A Smart contract is a set of promises, specified in digital form, including protocols within which the parties perform on these promises."

Source:
http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html

Creation of a Smart Contract - Ethereum

Creation of the Smart Contract

Regulators may also look to the formation of smart contracts as an opportunity to provide consumer protection. Regulators could require parties to hard-code certain terms or regulatory conditions into smart contracts as an enforcement tool. As an example, regulators could require parties to loans to input maximum interest rates to prevent usury and monitor for compliance. Coding requirements could lead to conflicts-of-law problems for companies who must answer to multiple federal regulators and the rules of several states (such as usury), especially when regulations change.

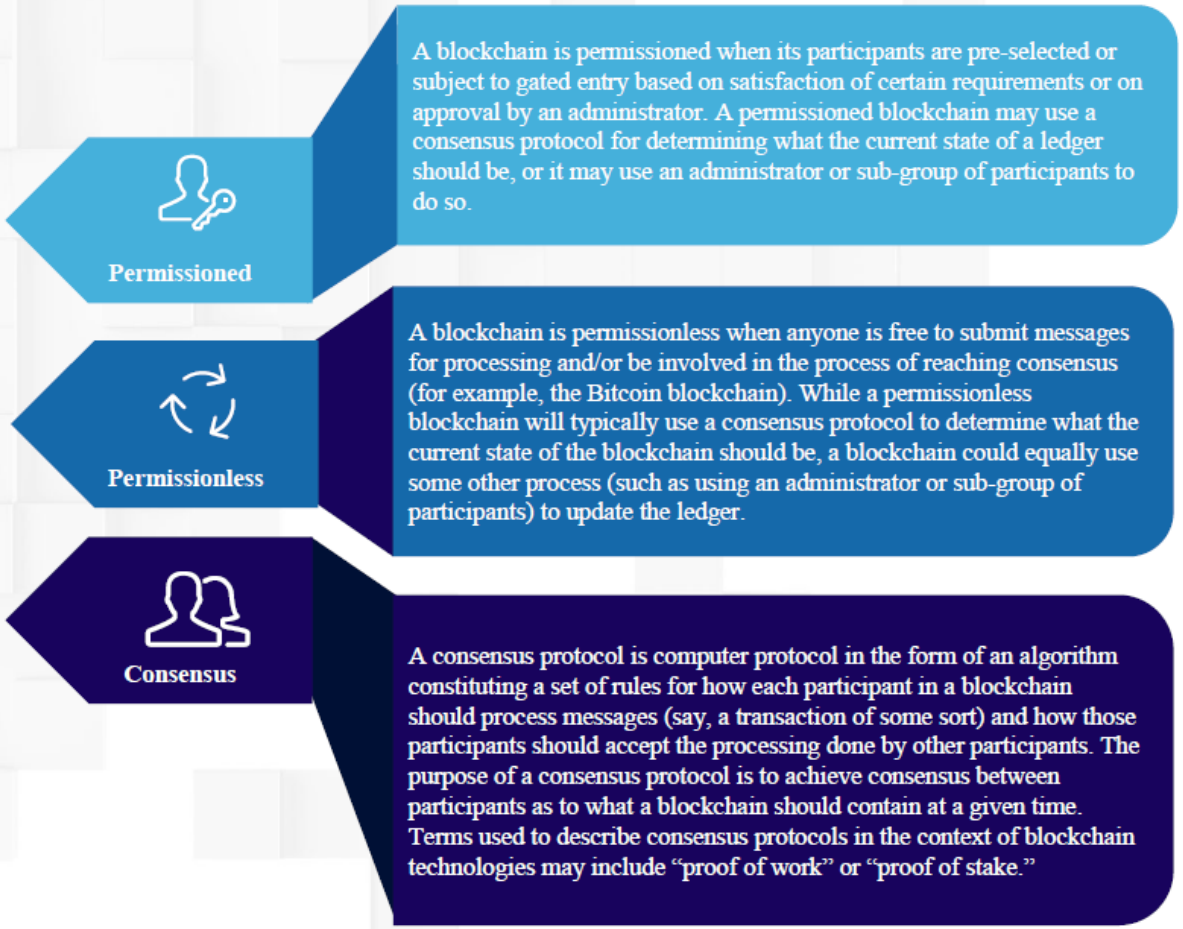
Considerations for Smart Contract Creation

- Notice of Terms to Parties
  - Visibility (are terms conspicuous?)
  - Timing (were terms shared before or after agreement?)
  - Difficulty (how hard must consumer work to see terms?)
- Sophistication of Parties
- Level of Control over Electronic Agents

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

How Is a Typical Smart Contract Initiated?

How is a typical smart contract initiated? It is necessary to have some understanding of the terminology:

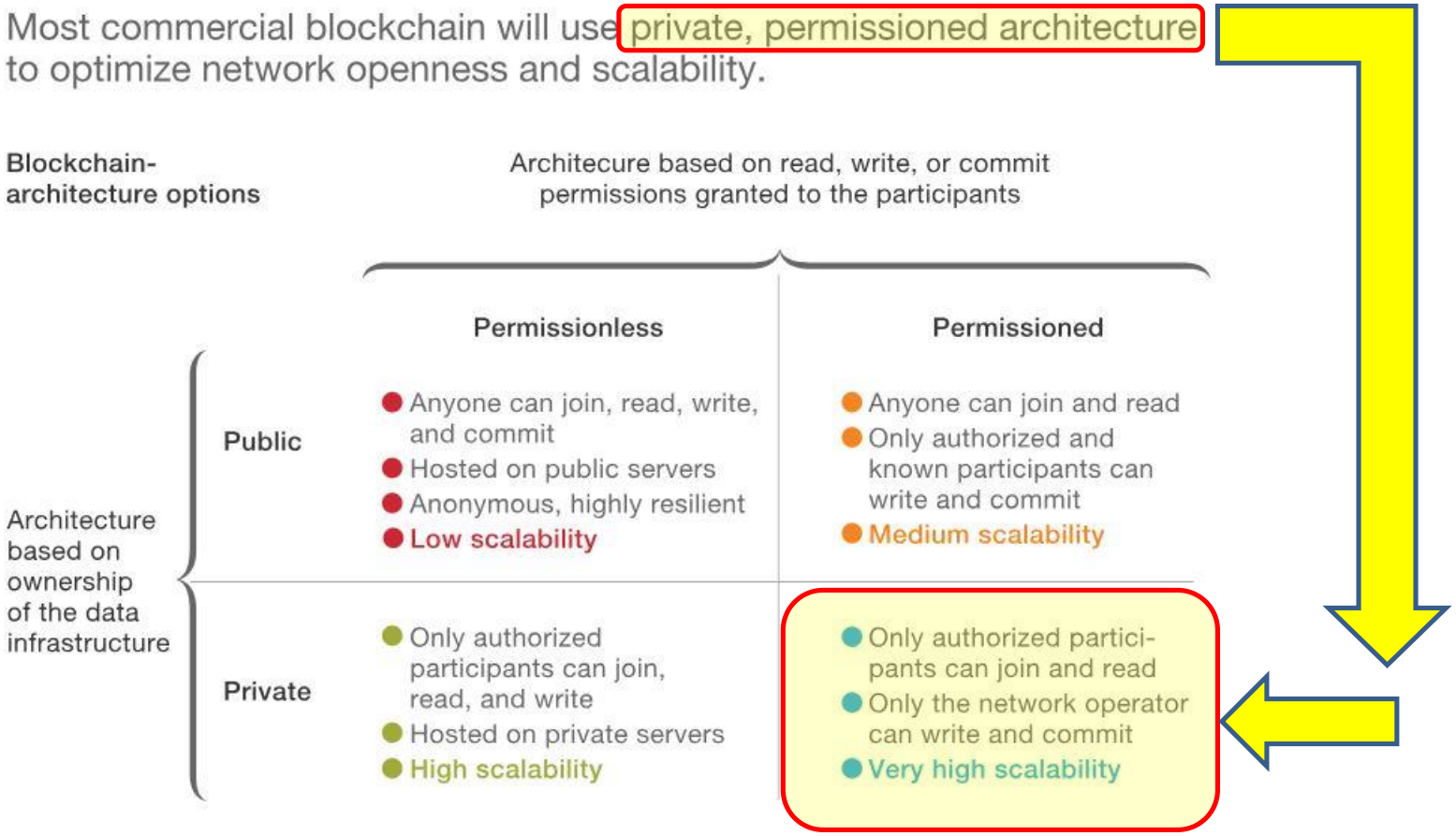


Source: Digital Chamber of Commerce
https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

Blockchain Architecture Examples

Exhibit 3

Most commercial blockchain will use private, permissioned architecture to optimize network openness and scalability.



McKinsey&Company

Formal Smart Contract Design – 6 Parts

1. Identity Management
2. Set Conditions
3. Code the Business Logic
   a) State Variables
   b) Functions
   c) Modifiers
   d) Events
4. Encryption and Blockchain Technology
5. Execution and Processing
6. Network Updates

Don't forget to Test! Test! Test!

Source: Digital Chamber of Commerce
https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

Different Models for Smart Contracts

What are the different models for smart contracts?

It is a common misconception that there is only one type of smart contract. In fact, there is a spectrum of possibilities.

Smart Contracts Lie on a Spectrum			
Contract entirely in code	Contract in code with separate natural language version	"Split" natural language contract with encoded performance	Natural language contract with encoded payment mechanism

Encoding Natural Language → Automation

Other permutations are, of course, possible and are likely to emerge as smart contract applications develop.

The role of code

The legal status of smart contracts is dealt with elsewhere in this white paper. For now, it is sufficient to note that smart contracts that seek to encode the entirety of a natural language contract (a "code is the contract" model) are very challenging from a legal perspective. The model puts into question an issue potentially relevant for all smart contracts: has a legally binding contract formed?

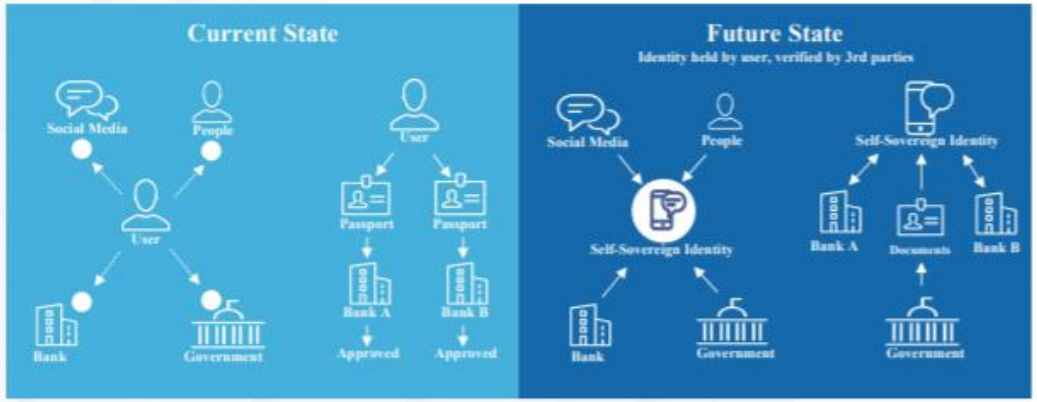
Source: Digital Chamber of Commerce
https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

Smart Contracts: 12 General Use Cases

1. Smart Contracts for Digital Identity

Smart Contracts for Digital Identity

Self-sovereign digital identity enabled by smart contracts provides seamless, user-centered internet for individuals.



Current Challenges

- Expensive and time consuming Know Your Customer (KYC) processes that lack completeness
- Limited control over potential data leakage due to an individual's reliance on trusted third-parties
- High liability to safeguard user data presents a single point-of-failure and a target for hackers

Smart Contract Benefits

- Individuals own and control personal data (e.g. able to securely disclose personal data to various counterparties)
- Counterparties will not need to hold sensitive data to verify transactions, reducing liability while facilitating frictionless KYC
- Increased compliance, resiliency and interoperability

Smart Contract Considerations

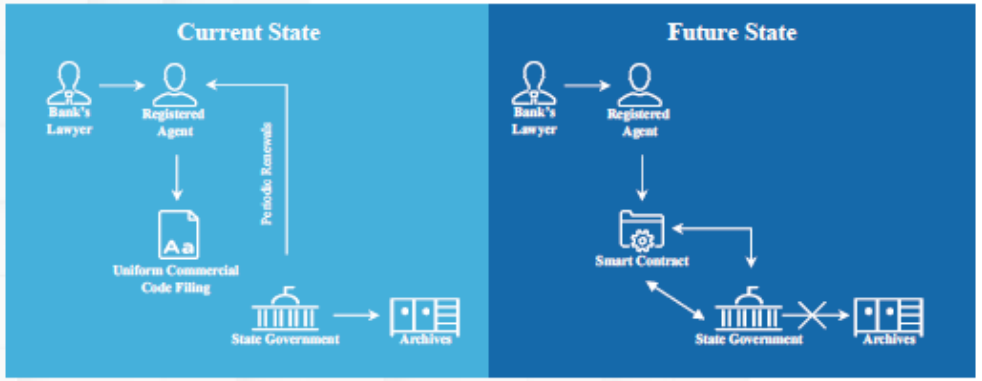
- Fostering an acceptance of digitally provided attestations within a legal framework and establishing trust in the security of smart contracts
- Technical integration with attestation providers
- Formation of protocols and standards to deliver interoperability by the involved parties

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

2. Smart Contracts for Records

Smart Contracts for Records

Automation of compliance, with rules requiring destruction of records on a future date enabled by smart contracts, and Uniform Commercial Code (UCC) liens that auto-renew, auto-release, or automatically call for collateral are all possible through smart contracts.



Current Challenges

- Paper-based filing for many foundational documents of finance with government
- Error-prone, manual process for renewing/releasing Uniform Commercial Code filings results in latency
- Expired archival data stored with government occupies warehouses and incurs additional costs

Smart Contract Benefits

- Reduced legal bills through auto-renewal and auto-release of digitized UCC filings
- Automated processes, including calling by lenders for additional collateral and tracking of loan vs. collateral value
- Archival data automatically becomes unsearchable/unreplayable after it reaches its sunset date

Smart Contract Considerations

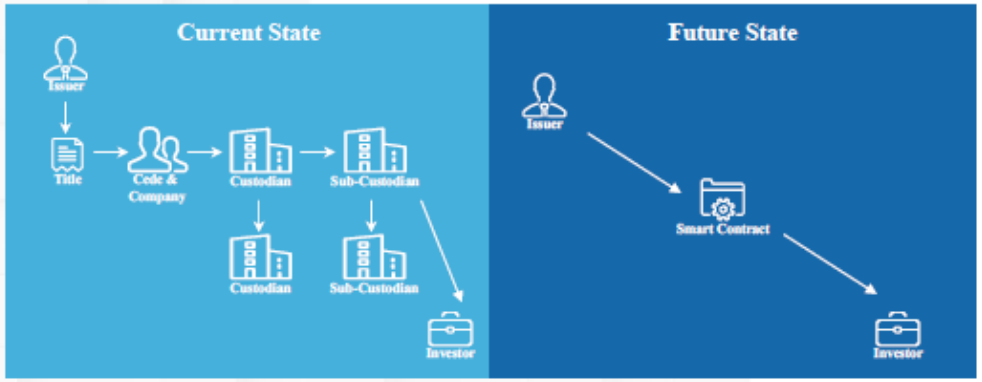
- Smart contract platform must be capable of storing data on a distributed ledger without slowing performance or compromising data privacy
- Active involvement of lenders and registered agents must exist for more complex functions (e.g. auto-release or automated call for additional collateral)
- Clarification regarding whether courts would consider a document legally destroyed if it is merely cryptographically unsearchable rather than removed from the ledger

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

3. Smart Contracts for Securities

Smart Contracts for Securities

Simplification of capitalization table management for private companies can be enabled by smart contracts, while also reuniting record ownership with beneficial ownership of publicly traded securities, reducing cost, and counterparty risk.



Current Challenges

- Paper-based, manual corporate registration processes
- Companies that fail to keep their corporate registrations up-to-date require clean-up and certificate of good standing before issuing securities
- Intermediaries increase cost, counterparty risk and latency

Smart Contract Benefits

- Digitized end-to-end workflows due to securities existing on a distributed ledger
- Trade date plus zero days (T+0) securities settlement cycles
- Facilitates automatic payment of dividends and stock splits, while enabling more accurate proxy voting
- Removes counterparty and operational risks created by intermediaries

Smart Contract Considerations

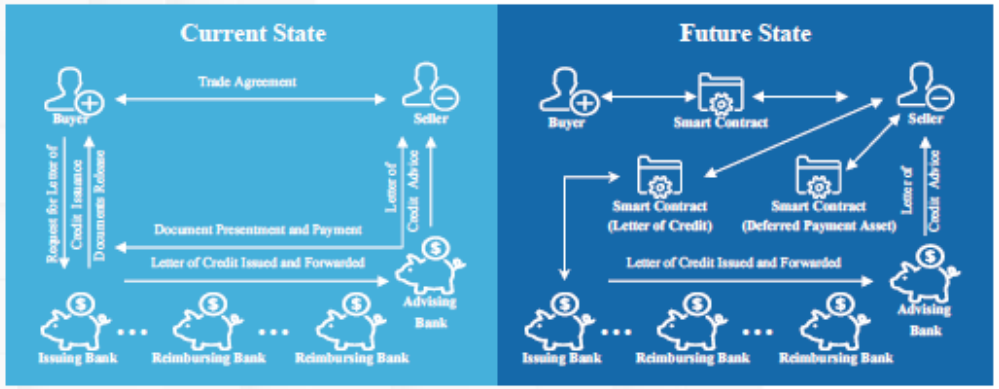
- Benefits may be realized more quickly in private securities markets than in public securities markets
- The cryptographic signature of the State of Delaware on the ledger entry takes the place of the State's seal on paper stock certificates, which may require enabling legislation to clarify that Delaware corporate law permits registration on a distributed ledger
- While issuers would welcome visibility into who owns their securities, some buy-side firms (e.g. activist investors) carefully protect this information

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

4. Smart Contracts for Trade Finance

Smart Contracts for Trade Finance

Payment method and instrument automation enabled by smart contracts provides risk mitigation and improved financing and process efficiencies for buyers, suppliers and financial institutions.



Current Challenges

- Time-consuming and costly Letter of Credit issuance process due to required coordination and paperwork
- Physical document management can delay shipment receipt until title document is released
- High document fraud/duplicate financing due to de-linked processes

Smart Contract Benefits

- Faster approval and payment initiation through automated compliance and monitoring of Letter of Credit conditions
- Improved efficiency in creating, modifying and validating trade, title and transport-related contract agreements
- Increased liquidity of financial assets due to ease of transfer and fraud reduction

Smart Contract Considerations

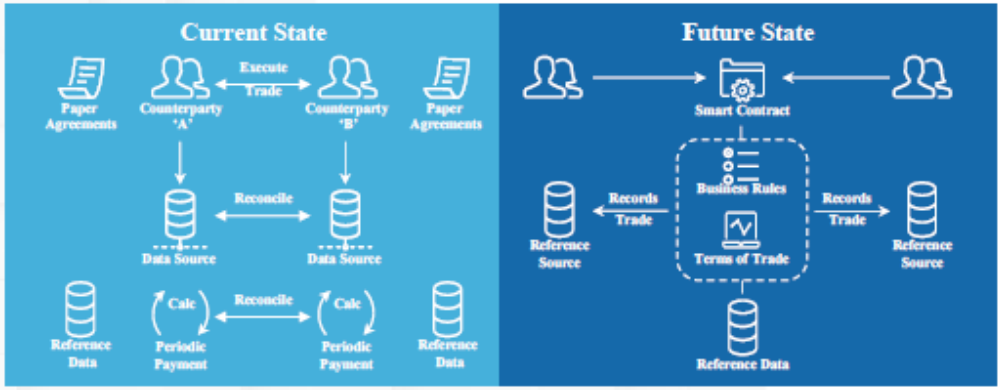
- Industry-wide standards for smart contract templates and procedures must be implemented for wider acceptability and adoption
- Legal implications for potential smart contract execution fall-out must be determined (in particular for defaults and dispute resolution)
- Integration with settlement systems, off-chain ecosystems and technology prerequisites (e.g. Internet of Things) must be successful to achieve full benefits

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

5. Smart Contracts for Derivatives

Smart Contracts for Derivatives

Enforcing a standard set of rules and conditions to a transaction enabled by smart contracts optimizes post-trade processing of over-the-counter (OTC) derivatives.



Current Challenges

- Redundant and time-consuming processes due to asset servicing being managed independently by each counterparty for most OTC derivatives
- Paper-based transaction agreements that contain terms, trade agreements and/or post-trade confirmations

Smart Contract Benefits

- Automated settlement of obligations while executing triggered processing of trade events (e.g. periodic payments)
- Automated external event processing (e.g. credit) and/or succession events
- Enabled real-time valuation of positions for real-time exposure monitoring, while reducing errors and/or disputes

Smart Contract Considerations

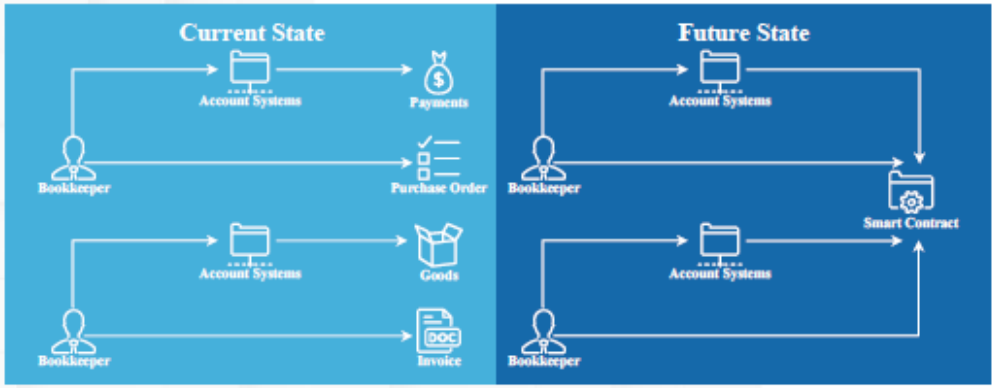
- Establish proper governance of a blockchain network and its smart contracts to properly manage large-scale protocol changes to existing contracts due to regulatory reform, change in contract or other unforeseen events
- Agreement upon lifecycle events for OTC derivatives (e.g. external source of data)
- Integration and governance of oracles required to feed smart contracts with information to/from the blockchain network

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

6. Smart Contracts for Financial Data Recording

Smart Contracts for Financial Data Recording

Smart contracts enable accurate recording of financial data for entities entering into financial transactions.



Current Challenges

- Accounting systems are prone to fraud and errors since they are controlled directly by entities
- Capital intensive processes due to each firm maintaining their own infrastructure
- Significant human capital/middleware required to process transactions from systems that do not interoperate

Smart Contract Benefits

- Improved transactional data integrity and transparency, yielding increased market stability
- Reduced expenditure for accounting information systems by cost-sharing across multiple organizations
- Improved insight into parties' capital due to increased financial accessibility

Smart Contract Considerations

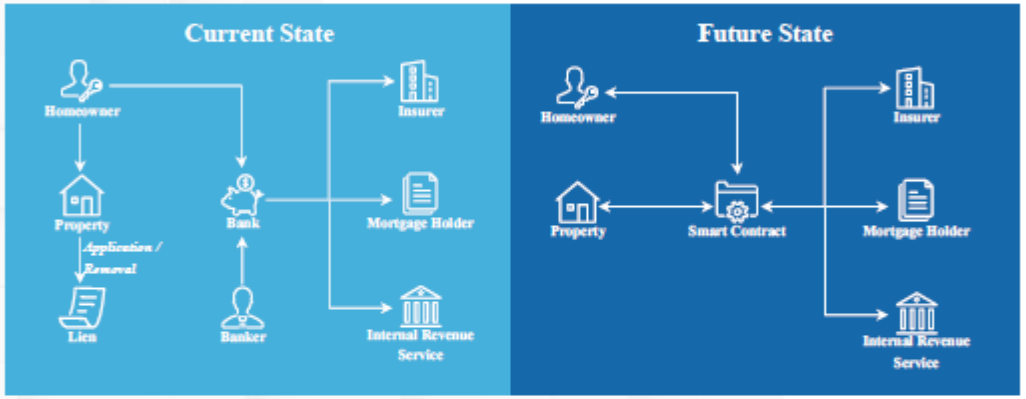
- Development of a portal to streamline smart contracts that facilitate and report financial transactions
- Design a set of standards for tokenized assets
- Interoperability between a distributed ledger network and legacy systems
- Creation of a marketplace of attesters to audit financial smart contracts

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

7. Smart Contracts for Mortgages

Smart Contracts for Mortgages

Mortgages enabled by smart contracts provide automated processing of payments and release of liens on property.



Current Challenges

- Process friction includes: payment application, updating balances, disbursing payments and taxes, and releasing liens when a mortgage is paid off
- Interface with auxiliary and dependent processes (e.g. land records)
- Privacy concerns due to security holders' needing to know borrowers' identities

Smart Contract Benefits

- Automated release of liens from land records when mortgage is paid off
- Increased visibility of servicer records to all interested parties, enabling payment verification and tracking
- Reduced cost and errors by elimination of manual processes

Smart Contract Considerations

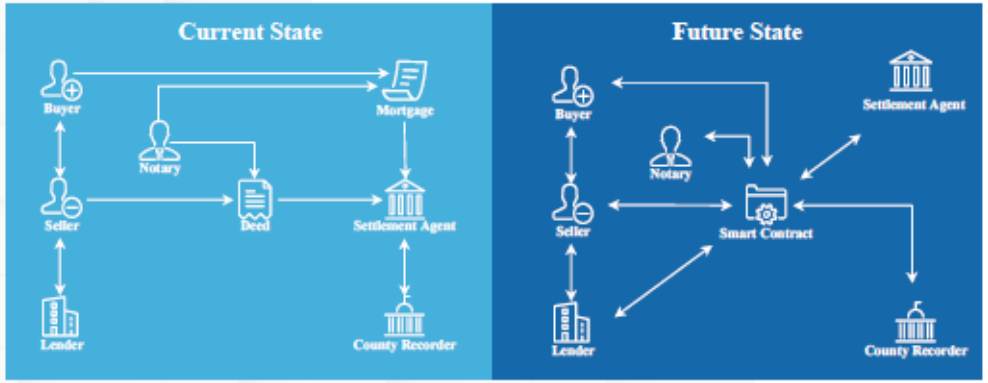
- Development of an interface between contract, borrower payment account, disbursement accounts and real estate title record service
- Digital identity must be successfully implemented to enable this use case
- Adoption of public key infrastructure between a mortgagee and the many parties involved

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

8. Smart Contracts for Land Title Recording

Smart Contracts for Land Title Recording

Property transfers enabled by smart contracts can deter fraud and improve transaction integrity, efficiency and transparency.



Current Challenges

- Capital intensity due to incompatible infrastructure
- Inefficient identity verification and signing process for documents
- Manual processes delay closing, escrow and recording processes and create potential for document alteration or loss
- Multiple parties can be shown the same property without detection

Smart Contract Benefits

- Higher confidence in identity of parties, streamlined processes and reduction in auditing/assurance costs
- Automated process notifications and incorporation of record integrity protections
- Reduce land title fraud conveyance
- Enhanced liquidity

Smart Contract Considerations

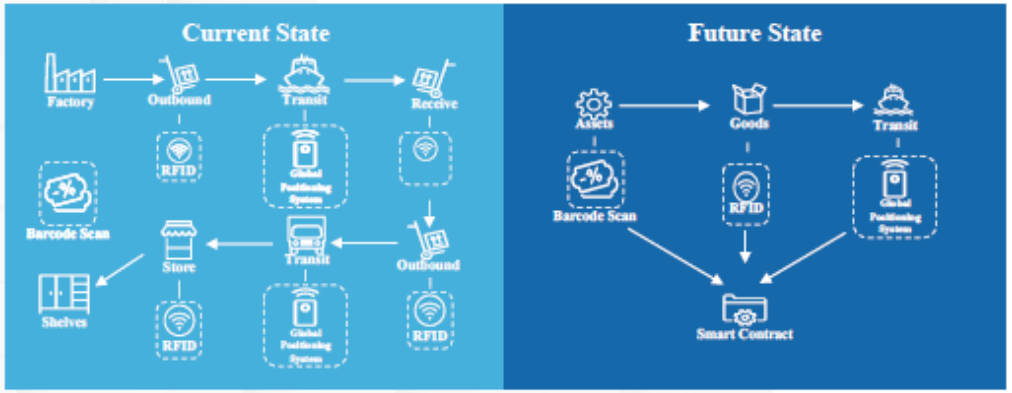
- Standardized record format (such as data elements and electronic signature fields) must be used by participating entities for deeds
- Common protocols must be developed for communication with all parties and electronic recording file formats
- Federated identity credentials must be accepted

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

9. Smart Contracts for Supply Chain

Smart Contracts for Supply Chain

Extended supply chain visibility, enabled by smart contracts, provides stand-up and tear-down of goods tracking across brands, retailers, logistics and contracted counterparties.



Current Challenges

- Limited visibility due to siloed data capture and desire to only share information with relevant parties
- Need for captured data to be similarly formatted to extract values
- Incompatibilities in data and blind spots in tracking goods due to silos in the supply chain (even source-tagged goods)

Smart Contract Benefits

- Simplification of complex multi-party systems delivery
- Achieve granular-level inventory tracking and delivery assurance, potentially improving supply chain financing, insurance and risk
- Enhanced tracing and verification to reduce risk of fraud and theft

Smart Contract Considerations

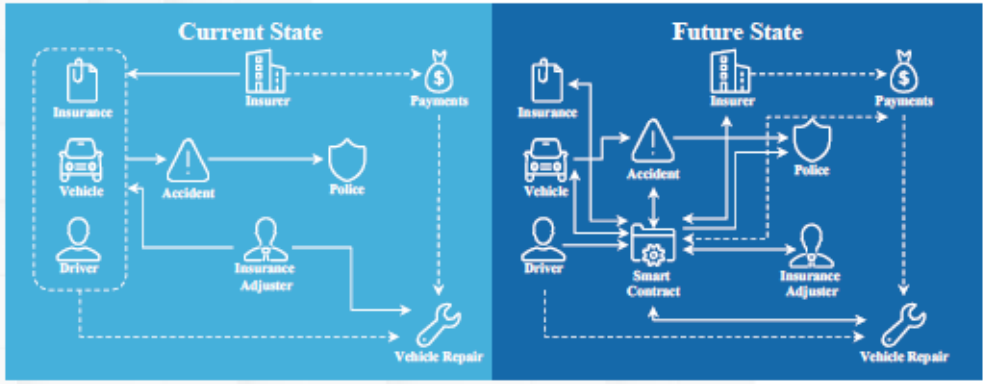
- Trusted oracles must be implemented to provide validated registrations of an entity
- Identities must be registered and attested over time, including for institutions, individuals, sensors, facilities and goods

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

10. Smart Contracts for Auto Insurance

Smart Contracts for Auto Insurance

Automated insurance claims enabled by smart contracts provide instantaneous processing, verification and payment by vehicles that are able to communicate with each other and assess and validate their own condition.



Current Challenges

- Multiple forms, reports and data sources yield increased error propensity and wasted time/resources
- Duplicated work due to insurance provider devoting back-office resources to verify records, reports and policies
- Subjective diagnostics during processes increases costs and delays

Smart Contract Benefits

- Repository for each policy holder includes global driving record, policy, vehicle type and accident report history
- Vehicle "self-awareness" and damage assessment using sensors can execute initial insurance claims/police reports
- Increased savings by reducing duplicated work to verify reports and policies

Smart Contract Considerations

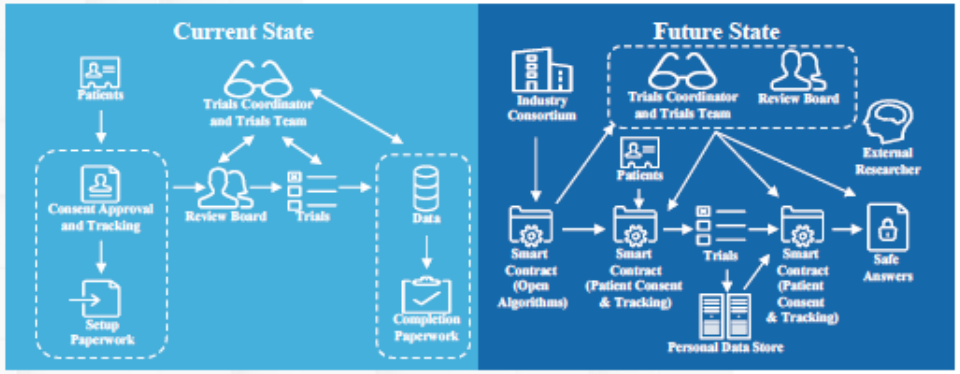
- Distributed Autonomous Policy (DAP) for ride-sharing companies that use contractors' cars and labor could be implemented, representing bundled, scalable and self-executing policies based on a driver's record, vehicle type and performance
- Innovation, cross-industry collaboration, and an environment open to testing and failing must be achieved to navigate the technological, financial and regulatory challenges

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

11. Smart Contracts for Clinical Trials

Smart Contracts for Clinical Trials

Increased visibility enabled by smart contracts may streamline the clinical trials process by increasing the sharing of data for participants in the ecosystem.



Current Challenges

- Delays in responding to epidemics due to friction in sharing data from clinical trials
- Limited understanding of treatment harms/benefits due to under-reporting
- Limited patient involvement due to lack of consistent consent management
- Comprisable patient privacy and re-identification due to sharing datasets

Smart Contract Benefits

- Increased visibility and reduced costs by streamlining setup processes for trials
- Improved access to cross-institution data during epidemics, protected by privacy-preserving computation
- Increased automation in obtaining and tracking consent for shared data access
- Increased confidence in patient privacy

Smart Contract Considerations

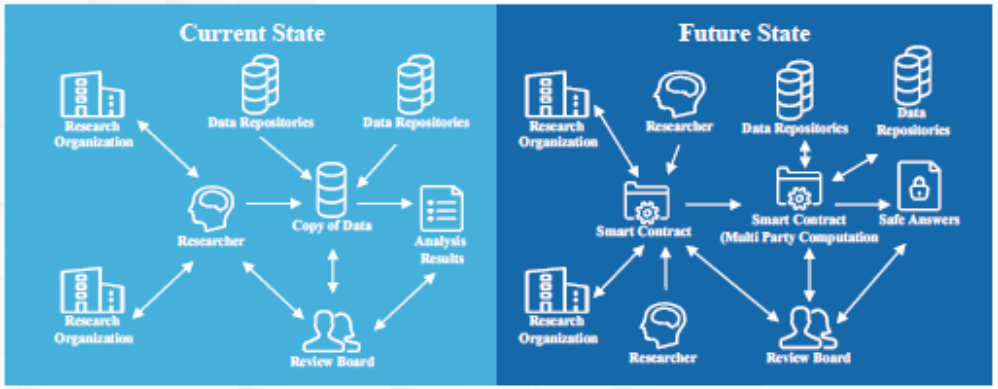
- Potential to cause positive disruption in the clinical trials community by providing scale to privacy-preserving data-sharing techniques and new multi-party computation architectures
- Identity, authentication and authorization remain open issues for smart contracts executable on blockchain enabled networks
- Potential path forward for the evolution of new data markets (e.g. clinical trials data market) based on new economic incentives models

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

12. Smart Contracts for Cancer Research

Smart Contracts for Cancer Research

Unleashed power of data enabled by smart contracts provides more efficient data sharing across sectors and incentivizes pre-competitive collaborations.



Current Challenges

- Cumbersome processes for sharing research across institutions
- Discouraged sharing of research due to privacy concerns
- Hindered data collection due to lack of trust and real-time access to patient data
- Deterred data sharing due to concerns around misaligned incentives

Smart Contract Benefits

- Enhanced data sharing while observing patient privacy/regulatory requirements
- Real-time visibility and policy enforcement incentivizes sharing without divulging raw data
- Increased volume of data and trust due to smart contract patient consent management

Smart Contract Considerations

- Standardization of privacy-safe queries and their representation in smart contracts must occur before benefits can be realized
- Transparency into allowable queries and available datasets backed by "open algorithms" that are vetted by experts must exist to ensure confidentiality
- Real-time access and protection of data confidentiality may require development of new forms of blockchain technologies

Source: Digital Chamber of Commerce https://digitalchamber.org/wp-content/uploads/2018/02/Smart-Contracts-12-Use-Cases-for-Business-and-Beyond_Chamber-of-Digital-Commerce.pdf

Case Studies

Case Study 01

- Timeframe: Summer of 2018
- Location: Chicago
- Topic: Teaching Interns who were Technical People with Graduate degrees free Blockchain classes
- 33 started, only 4 remain
- First Project: We are converting and existing Time Tracking GUI Application to an Ethereum DApp
- Second Project: Designing and Implementing a DApp Solution from Scratch
- We worked together from June 1 – December 31, 2018

- What happened?
  - Lazy, uncommitted interns
  - Deception and fraud with USCIS (they were there to fraudulently extend their stay in the U.S.)

Case Study 02

- Timeframe: November 2017
- Location: User ***devops199*** somewhere on the Ethereum Blockchain
- Topic: Placement in Production of flawed Smart Contract
- Results: Loss of over $150 million

Case 02 – Horror Story – the $150 Million Bug

9 ■■■■ js/src/contracts/snippets/enhanced-wallet.sol — Show comments — View

```
@ -104,7 +104,7 @@ contract WalletLibrary is WalletEvents {
104
105    // constructor is given number of sigs required to do protected "onlymanyowners" transactions
106    // as well as the selection of addresses capable of confirming them.
107 -  function initMultiowned(address[] _owners, uint _required) {
108      m_numOwners = _owners.length + 1;
109      m_owners[1] = uint(msg.sender);
110      m_ownerIndex[uint(msg.sender)] = 1;
@ -198,7 +198,7 @@ contract WalletLibrary is WalletEvents {
198  }
199
200  // constructor - stores initial daily limit and records the present day's index.
201 -  function initDaylimit(uint _limit) {
202    m_dailyLimit = _limit;
203    m_lastDay = today();
204  }
@ -211,9 +211,12 @@ contract WalletLibrary is WalletEvents {
211    m_spentToday = 0;
212  }
213
```

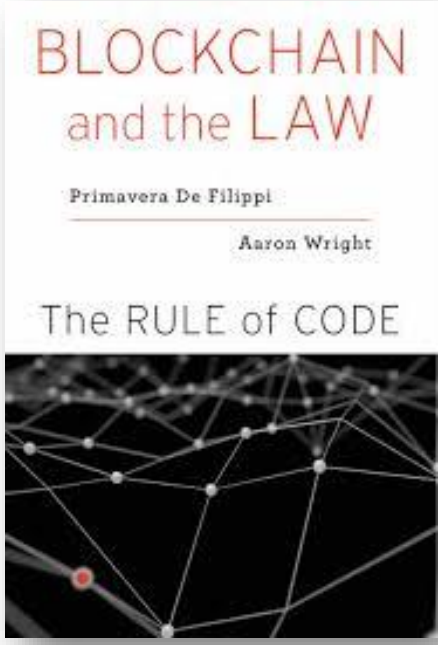
```
@ -104,7 +104,7 @@ contract WalletLibrary is WalletEvents {
104
105    // constructor is given number of sigs required to do protected "onlymanyowners" transactions
106    // as well as the selection of addresses capable of confirming them.
107 +  function initMultiowned(address[] _owners, uint _required) internal {
108      m_numOwners = _owners.length + 1;
109      m_owners[1] = uint(msg.sender);
110      m_ownerIndex[uint(msg.sender)] = 1;
@ -198,7 +198,7 @@ contract WalletLibrary is WalletEvents {
198  }
199
200  // constructor - stores initial daily limit and records the present day's index.
201 +  function initDaylimit(uint _limit) internal {
202    m_dailyLimit = _limit;
203    m_lastDay = today();
204  }
@ -211,9 +211,12 @@ contract WalletLibrary is WalletEvents {
211    m_spentToday = 0;
212  }
213
214 +  // throw unless the contract is not yet initialized.
215 +  modifier only_uninitialized { if (m_numOwners > 0) throw; _; }
```

Blockchain Law

Blockchain and the Law

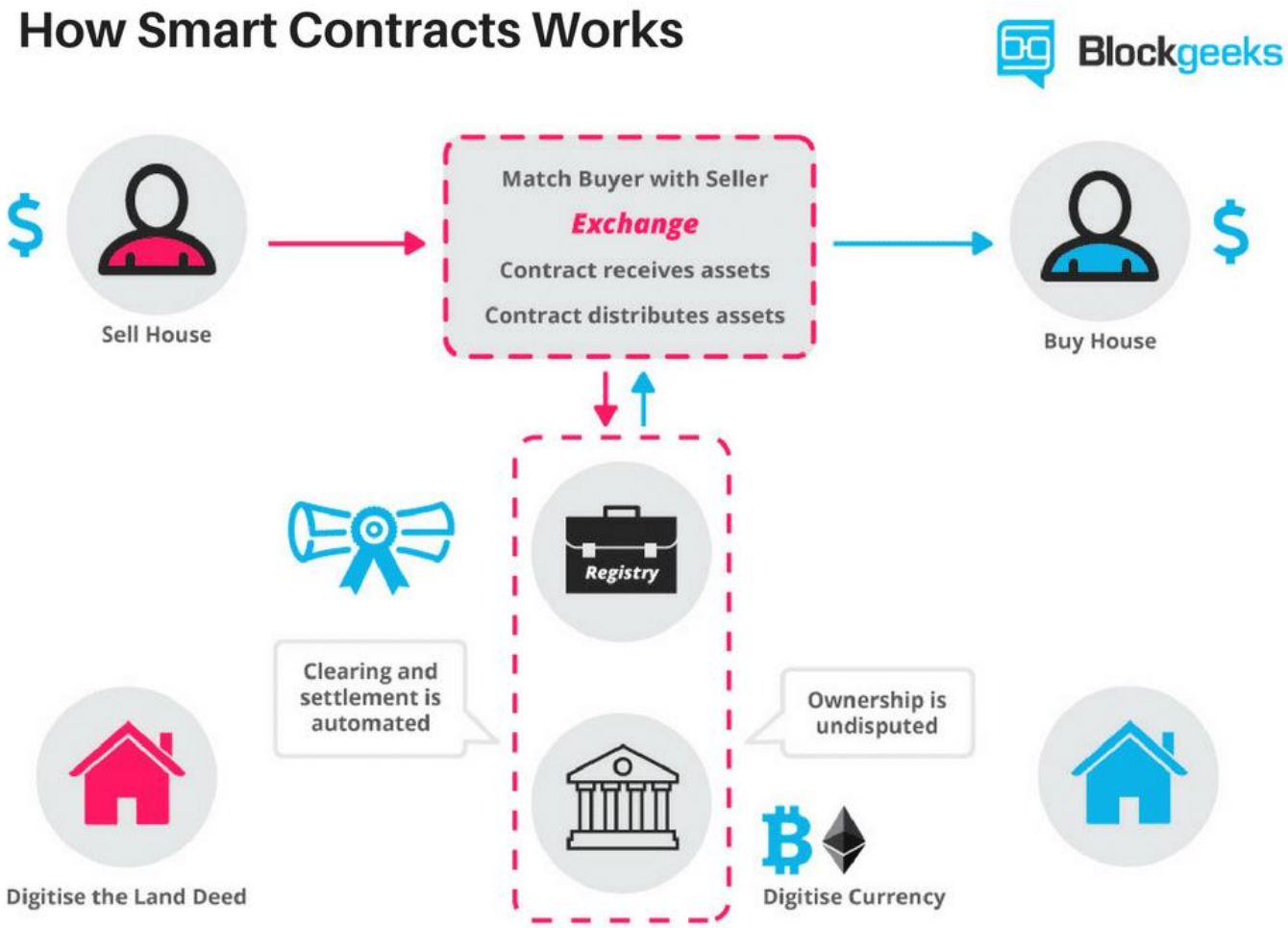
- Blockchain establishes ownership, confirmed transactions, control, and transfer of ownership.
- Blockchain will force lawyers to understand technology better
- Blockchain could also make room for “smart contracts,” where assets would be transferred automatically once certain conditions are met.
- Blockchain could resolve disputes very directly and efficiently, saving lawyers and their clients a great deal of work. This also could mean the end of escrow accounts where the law firm holds onto money and distributes funds once conditions have been met.
- Contracts and transactions could be a logical first-step in the blockchain adoption journey.
- Blockchain could very well improve the effectiveness of the criminal justice system;
- If corporations and websites agree to give law firms access to records automatically collected through blockchain, those records could cause new, reliable evidence to surface more quickly.
- Expect that those with evidence on their side will embrace this concept, and others will prefer to drag their adversary through a drawn-out process.
- As more companies adopt Blockchain technologies and require their third-party suppliers to adopt Blockchain technologies, expect this requirement to be written into legally binding business contracts.

Source: https://www.forbes.com/sites/ianaltman/2018/06/29/blockchain-changes-business-law/#698d3605cb9f



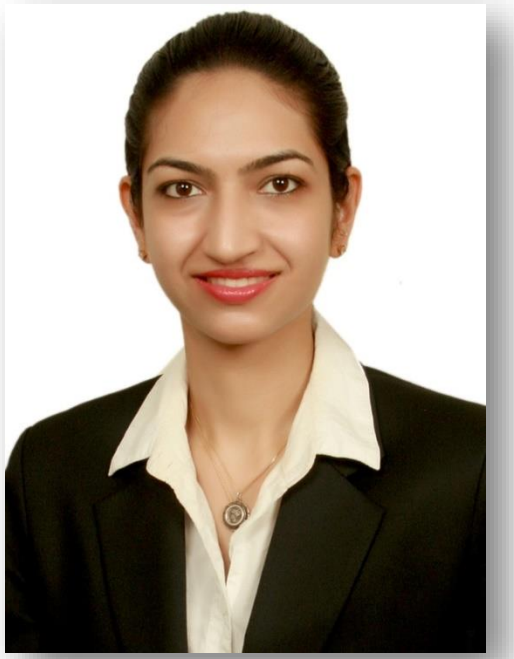
For more information
Get
Blockchain & the Law
By Primavera De Filippi
And Aaron Wright, 2018

Blockchain & the Law



Blockchain & the Law – Some Actual Legal Resources

Nelson Rosario
Chicago
https://www.linkedin.com/in/nelsonrosario/



Ms. Puneet Bhasin
Mumbai, India
https://www.linkedin.com/in/advpuneetbhasincyberlawyer/

Topic 7: Blockchain Limits and Challenges

Technical Limitations

The most important technical limitations of the blockchain are:

- Lack of privacy
- The security model
- Limited scalability
- High costs
- Hidden centrality
- Lack of flexibility
- Critical size

Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

Technical Limitations

Table 23-1. Technical Limitations of the Blockchain and Their Reasons

Technical Limitation	Conflict	Fundamental Functionality
Lack of privacy	Transparency vs. privacy	Reading the history of transaction data
Lack of scalability	Security vs. speed	Writing transaction data to the data store

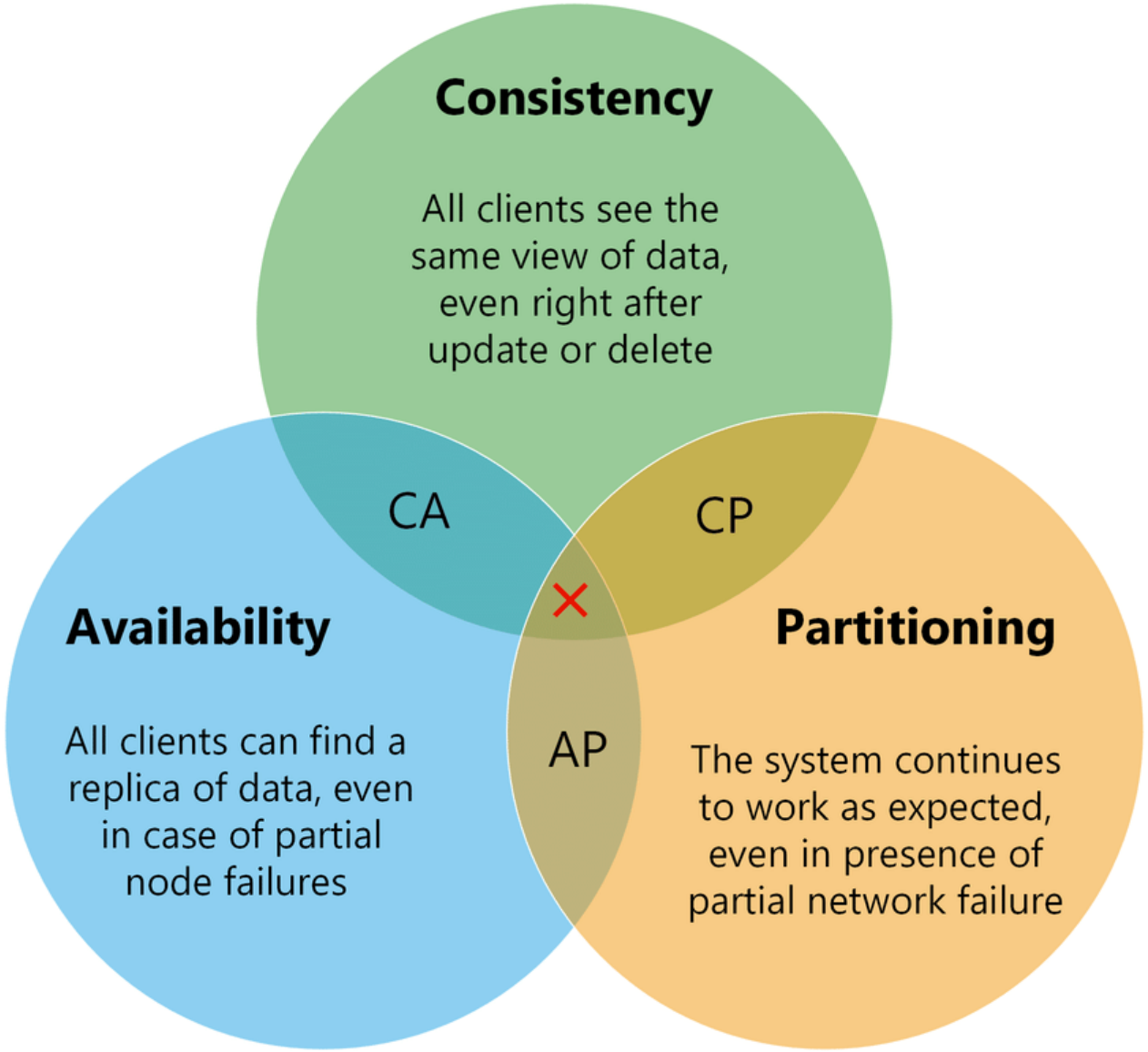
Source: Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

Technical Limits & Challenges

- Scalability
- Performance (Bitcoin – 600 seconds / block; Ethereum, 14 to 17 seconds / block)
- Security, especially with user wallets
- Weaknesses in the technologies, i.e. deployment of bad contracts, can cause very expensive blunders and loss of confidence and reputation
- Finding the right people to develop DApps and manage the technologies
- Resistance to change
- Anti-trust issues (Norton Rose Fulbright):
- Does blockchain allow for improper information sharing and facilitate collusion among competitors?
- Do blockchain standards and rules create or enhance market power by favoring one or several industry participant(s) over others?
- Does a permissioned blockchain amount to a concerted refusal to deal?

Distributed System Concepts

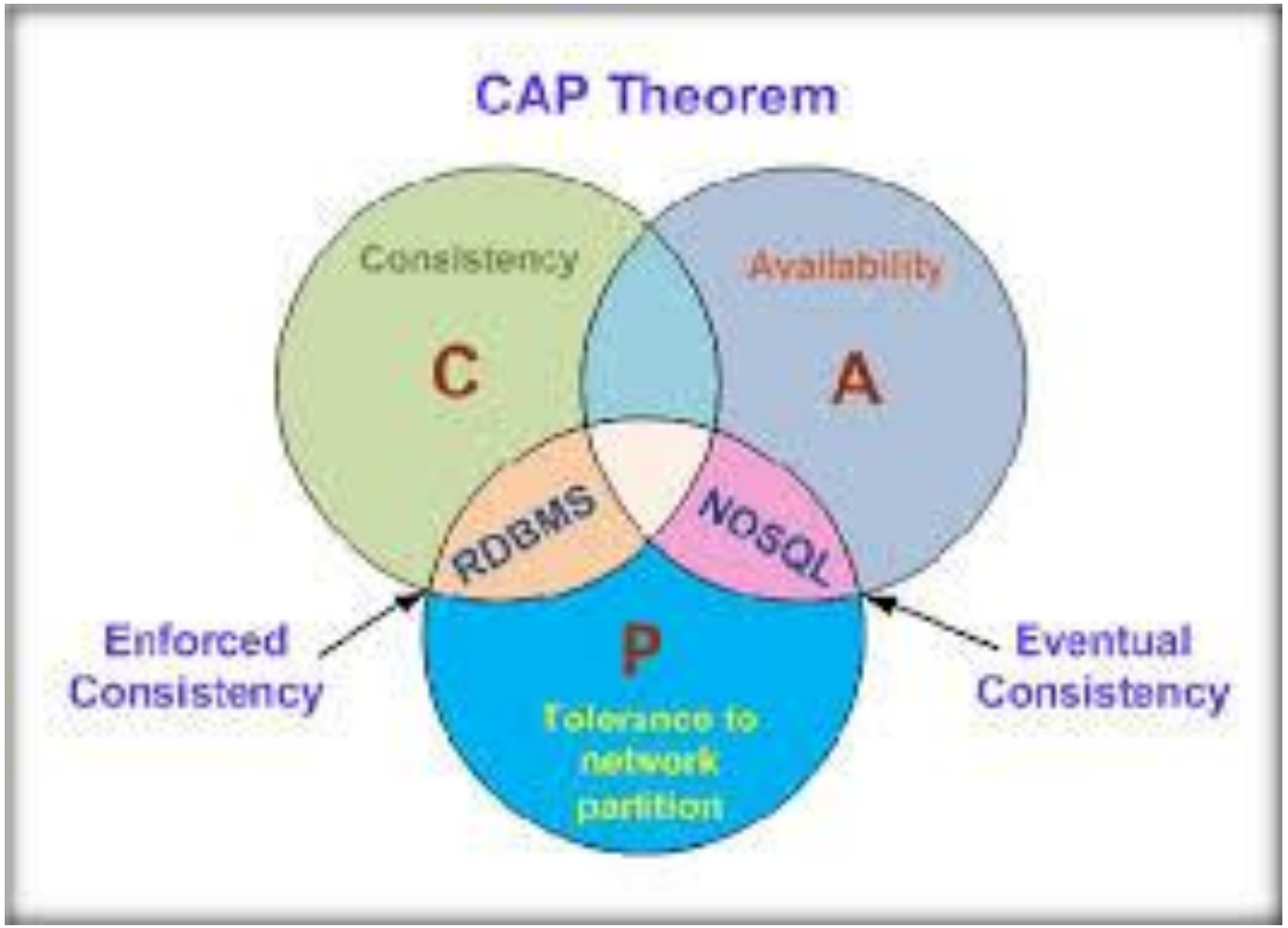
CAP Theorem



In a Distributed System, you can only design for and meet the requirements of TWO of these Characteristics.

Source: https://en.wikipedia.org/wiki/CAP_theorem

CAP Theorem



Source: https://en.wikipedia.org/wiki/CAP_theorem

CAP Theorem

CAP Theorem

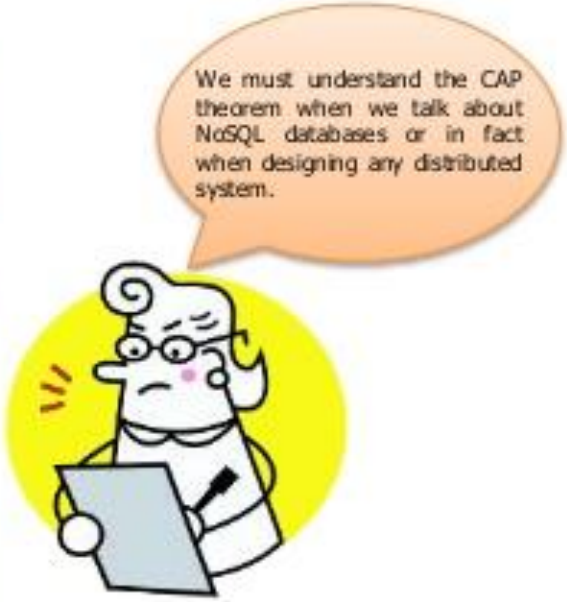
edureka!

CAP theorem states that there are **3 basic requirements** which exist in a special relation when designing applications for a distributed architecture.

Consistency — This means that the data in the database remains consistent after the execution of an operation. For example after an update operation all clients see the same data.

Availability — This means that the system is always on (service guarantee availability), no downtime.

Partition Tolerance — This means that the system continues to function even the communication among the servers is unreliable, i.e. the servers may be partitioned into multiple groups that cannot communicate with one another.



Slide 7

Twitter @edurekaIN, Facebook /edurekaIN, use #askEdureka for Questions

www.edureka.in

Source: Edureka.in

CAP Theorem

CAP Theorem

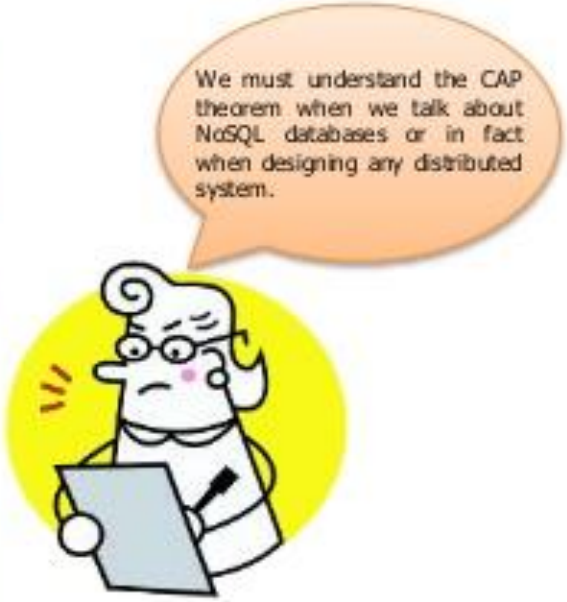
edureka!

CAP theorem states that there are **3 basic requirements** which exist in a special relation when designing applications for a distributed architecture.

Consistency — This means that the data in the database remains consistent after the execution of an operation. For example after an update operation all clients see the same data.

Availability — This means that the system is always on (service guarantee availability), no downtime.

Partition Tolerance — This means that the system continues to function even the communication among the servers is unreliable, i.e. the servers may be partitioned into multiple groups that cannot communicate with one another.



Slide 7

Twitter @edurekaIN, Facebook /edurekaIN, use #askEdureka for Questions

www.edureka.in

Source: Edureka.in

CAP Theorem

CAP theorem

From Wikipedia, the free encyclopedia

In theoretical computer science, the **CAP theorem**, also named **Brewer's theorem** after computer scientist Eric Brewer, states that it is impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees:[1][2][3]

- *Consistency*: Every read receives the most recent write or an error
- *Availability*: Every request receives a (non-error) response – without the guarantee that it contains the most recent write
- *Partition tolerance*: The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes

In particular, the CAP theorem implies that in the presence of a network partition, one has to choose between consistency and availability. Note that consistency as defined in the CAP theorem is quite different from the consistency guaranteed in ACID database transactions.

Source: https://en.wikipedia.org/wiki/CAP_theorem

CAP Theorem

Explanation [edit]

No distributed system is safe from network failures, thus network partitioning generally has to be tolerated. In the presence of a partition, one is then left with two options: consistency or availability. When choosing consistency over availability, the system will return an error or a time-out if particular information cannot be guaranteed to be up to date due to network partitioning. When choosing availability over consistency, the system will always process the query and try to return the most recent available version of the information, even if it cannot guarantee it is up to date due to network partitioning.

In the absence of network failure – that is, when the distributed system is running normally – both availability and consistency can be satisfied.

CAP is frequently misunderstood as if one has to choose to abandon one of the three guarantees at all times. In fact, the choice is really between consistency and availability only when a network partition or failure happens; at all other times, no trade-off has to be made.[4][5]

Database systems designed with traditional ACID guarantees in mind such as RDBMS choose consistency over availability, whereas systems designed around the BASE philosophy, common in the NoSQL movement for example, choose availability over consistency.[6]

The PACELC theorem builds on CAP by stating that even in the absence of partitioning, another trade-off between latency and consistency occurs.

History [edit]

According to University of California, Berkeley computer scientist Eric Brewer, the theorem first appeared in autumn 1998.[6] It was published as the CAP principle in 1999[7] and presented as a conjecture by Brewer at the 2000 Symposium on Principles of Distributed Computing (PODC).[8] In 2002, Seth Gilbert and Nancy Lynch of MIT published a formal proof of Brewer's conjecture, rendering it a theorem.[1]

In 2012, Brewer clarified some of his positions, including why the often-used "two out of three" concept can be misleading or misapplied, and the different definition of consistency used in CAP relative to the one used in ACID.[6]

A similar theorem stating the trade-off between consistency and availability in distributed systems was published by Birman and Friedman in 1996.[9] The result of Birman and Friedman restricted this lower bound to non-commuting operations.

Source: https://en.wikipedia.org/wiki/CAP_theorem

Topic 8: Blockchain Security

3 Important Things Business Leaders Needs to Know About Blockchain Security

1. Security is not just a technical problem, it is a leadership problem

2. Exploitation is not just a result of attacker capabilities, but also of developer errors

3. While attackers do compromise a blockchain itself, they more commonly exploit the configuration of the technology leveraging a blockchain

Source: Alison DeNisco Ramone, TechRepublic.com, April 18, 2019
https://www.techrepublic.com/article/how-to-secure-a-blockchain-3-things-business-leaders-need-to-know/

How to Secure Blockchain Applications and Infrastructure

- Build and lead Teams of experienced, dedicated workers
- Design securely
- Do code reviews and rigorous testing
- Implement securely
- Document ***everything***
- Test security
- Routinely test vulnerabilities (at least quarterly)
  - https://tinyurl.com/y292y3yf
- Penetration test semi-annually
  - https://tinyurl.com/yya4vtac
- Test and document performance
  - https://tinyurl.com/yxpwszj7
- Do Threat Management
- Continuously review for upgrading

Blockchain Security – Threats and Vulnerabilities & Remediation – Part 1

Threat or Vulnerability	Description	Remediation	Comment(s)
Threat	51% Attack	Securely design, implement, monitor, maintain, test & upgrade.	Happened to Bitcoin in June 2014. http://tinyurl.com/y5malrxc
Threat	Sybil Attack	Securely design, implement, monitor, maintain, test & upgrade.	Need better education and experience.
Vulnerability	Bad Private Key Management	Understand & Securely manage private keys.	Need better education and tools.
Vulnerability	Centralization	Understand the CAP Theorem and Decentralization. Design and implement accordingly.	Need better education.
Vulnerability	Scalability	Securely design, implement, monitor, maintain, test & upgrade.	Need better education and experience.
Vulnerability	Network Security	Securely design, implement, monitor, maintain, test & upgrade.	Need better education.
Vulnerability	Smart Contracts – Coding errors	Securely design, implement, monitor, maintain, test & upgrade.	Need better education and experience.
Vulnerability	Smart Contracts – Configuration Errors	Securely design, implement, monitor, maintain, test & upgrade.	Need better education and experience.
Vulnerability	Blockchain & Smart Contracts - Inexperience	Use Secure Development practices, and experienced developers and testers.	Need better education and experience.

Blockchain Security – Threats and Vulnerabilities & Remediation – Part 2

Threat or Vulnerability	Description	Remediation	Comment(s)
Vulnerability	Reentrancy	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	Unexpected Ether	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	DELEGATECALL	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	Default Visibilities	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	Entropy Illusion	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	External Contract Referencing	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	Short Address / Parameter Attack	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	Unchecked CALL Return Value	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	Race Conditions / Front Running	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.

Blockchain Security – Threats and Vulnerabilities & Remediation – Part 3

Threat or Vulnerability	Description	Remediation	Comment(s)
Vulnerability	Denial of Service	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	Block Timestamp Manipulation	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	Constructions with Care	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	Uninitialized Storage Pointers	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	Floating Point and Precision	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	Transaction Origin Authentication	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Vulnerability	Contract Libraries	Securely design, implement, monitor, maintain, test & upgrade. Code reviews & Audits.	See ***Mastering Ethereum***, Chapter 9.
Threat	**Shor's Algorithm (Using Quantum Computing)**	**Stronger, better encryption, perhaps Quantum Cryptography.**	**Closer than you think**

MIT Article – Blockchains Are Now Getting Hacked

51% Attack on Ethereum Classic – January 2019

Source: MIT Review, Mike Orcutt, February 19, 2019
https://www.technologyreview.com/s/612974/once-hailed-as-unhackable-blockchains-are-now-getting-hacked/

Once hailed as unhackable, blockchains are now getting hacked

More and more security holes are appearing in cryptocurrency and smart contract platforms, and some are fundamental to the way they were built.

by Mike Orcutt February 19, 2019

Early last month, the security team at Coinbase noticed something strange going on in Ethereum Classic, one of the cryptocurrencies people can buy and sell using Coinbase's popular exchange platform. Its blockchain, the history of all its transactions, was under attack.

An attacker had somehow gained control of more than half of the network's computing power and was using it to rewrite the transaction history. That made it possible to spend the same cryptocurrency more than once—known as "double spends." The attacker was spotted pulling this off to the tune of $1.1 million. Coinbase claims that no currency was actually stolen from any of its accounts. But a second popular exchange, Gate.io, has admitted it wasn't so lucky, losing around $200,000 to the attacker (who, strangely, returned half of it days later).

Just a year ago, this nightmare scenario was mostly theoretical. But the so-called 51% attack against Ethereum Classic was just the latest in a

How to Perform Secure Software Development for Blockchain Applications by Design, Coding practices, Testing and Verification

- Experienced DApp developers
- Test-driven Development
- Code Defensively
- Code reviews, by multiple experienced developers
- Understand and remediate the weakest security points, especially protection of private keys and sensitive data.
- Implement the tests on test net and understand exactly how the code will behave prior to moving to main net
- Automate Smart Contract testing when possible

Ethereum Smart Contract Security Best Practices

Ethereum Smart Contract Security Best Practices

This document provides a baseline knowledge of security considerations for intermediate Solidity programmers. It is maintained by ConsenSys Diligence, with contributions from our friends in the broader Ethereum community.

Where to start?

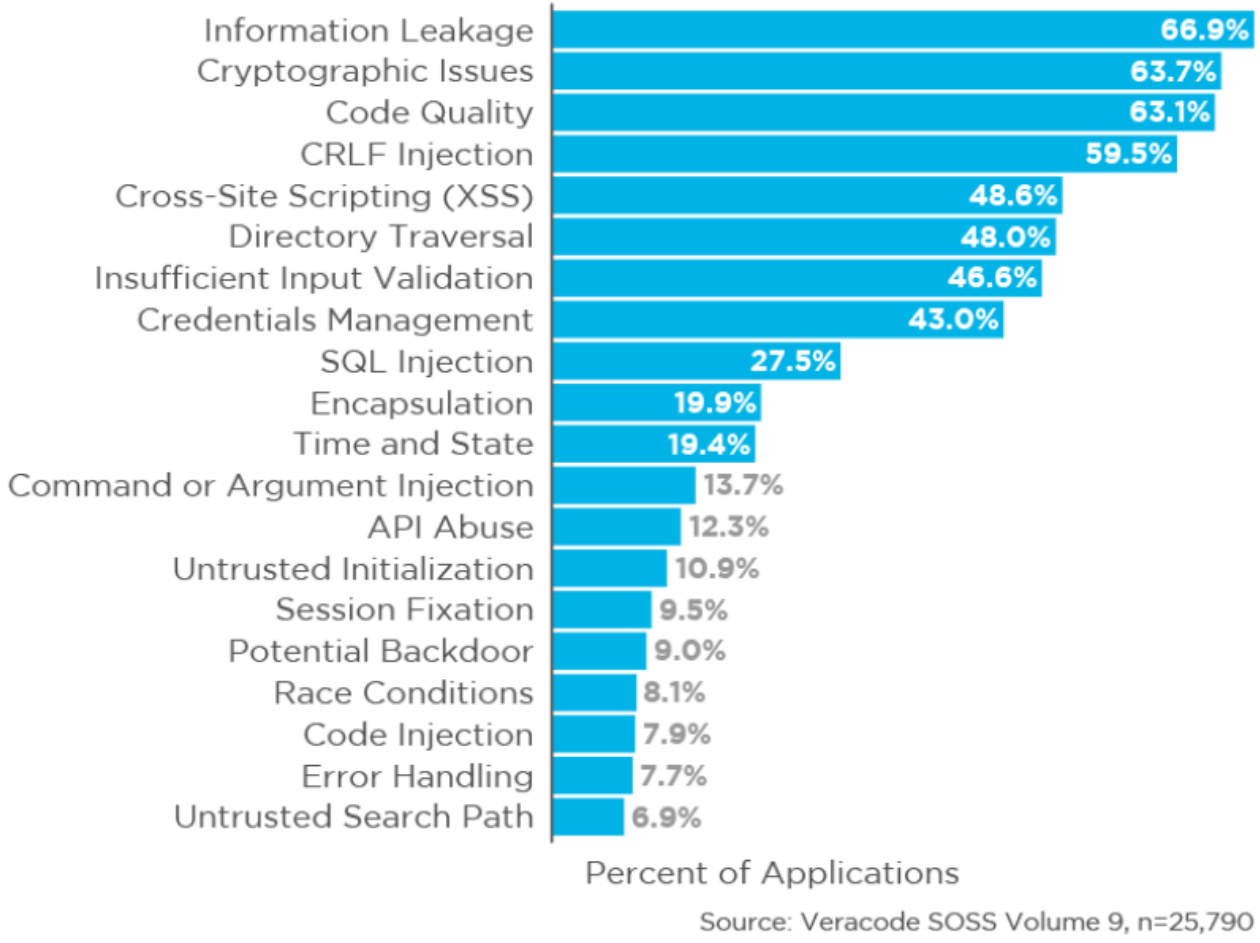
- General Philosophy describes the smart contract security mindset
- Solidity Recommendations contains examples of good code patterns
- Known Attacks describes the different classes of vulnerabilities to avoid
- Software Engineering outlines some architectural and design approaches for risk mitigation
- Documentation and Procedures outlines best practices for documenting your system for other developers and auditors
- Security Tools lists tools for improving code quality, and detecting vulnerabilities
- Security EIPs lists EIP's related to security issues and vulnerabilities
- Security Resources lists sources of information for staying up to date
- Tokens outlines best practices specifically related to Tokens.

Best Free Resources
On Smart Contract Security
Best Practices

Smart Contract Security Best Practices
https://consensys.github.io/smart-contract-best-practices/

Top Web Application & Software Vulnerabilities - The Story in 2019

FIGURE 23: 20 MOST COMMON VULNERABILITY CATEGORIES



Source: Veracode SOSS Volume 9, n=25,790

Source: https://www.veracode.com/sites/default/files/pdf/resources/ipapers/state-of-software-security-volume-9/index.html

Topic 9: Examples of Real-world Blockchain Applications

Real-World Blockchain Solutions

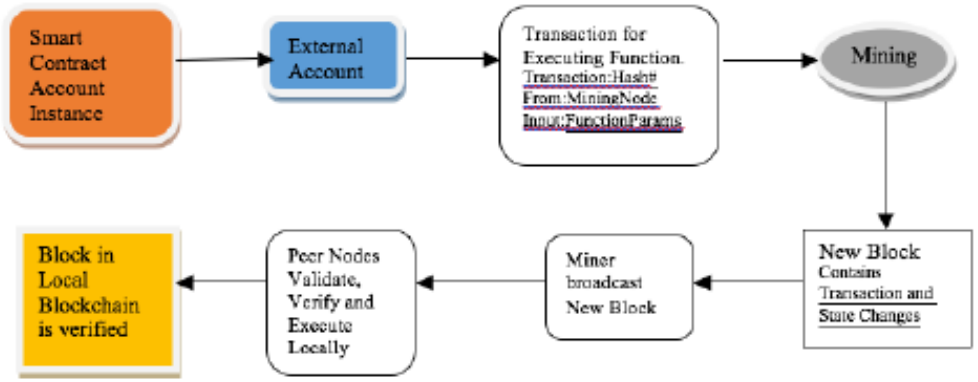
Entity	Use	Blockchain(s)	Link
Maersk	Expedite tracking of Cargo shipment internationally	Hyperledger	https://www.ibm.com/blogs/think/2018/11/tradelens-how-ibm-and-maersk-are-sharing-blockchain-to-build-a-global-trade-platform/
U.S. State Department & Coca-Cola	Reduce risk of forced labor and child labor	Customized	https://www.digitaltrends.com/cool-tech/coca-cola-blockchain-forced-labor/
Saudi Arabia	Tracking cross-border trade	Hyperledger	https://cointelegraph.com/news/saudi-arabia-completes-ibm-tradelens-pilot-for-cross-border-blockchain-trade
Overstock	Business model change from online retail to investor in Blockchain and Cryptocurrency Start-ups	Several	https://mashable.com/article/overstock-blockchain-cryptocurrency/
Walmart	Requiring several fresh food suppliers to use Blockchain	Several	https://cointelegraph.com/news/walmart-requires-certain-produce-suppliers-to-deploy-blockchain-technology
FedEx	Supply chain and logistics management improvements.	Hyperledger	https://cointelegraph.com/news/fedex-joins-hyperledger-blockchain-hub-big-implications-for-logistics

Topic 10: The Ethereum EVM, Smart Contracts, and Solidity

Smart Contract Execution

SMART CONTRACT EXECUTION

A Smart Contract contains functions that can be executed by an External Account or a Decentralized Application (DAPP). In the case of a DAPP, the executing node would have a default External Accoun



- To execute a function defined in the Smart Contract, the DAPP retrieves a unique instance of the Smart Contract by its address.
  E.g.

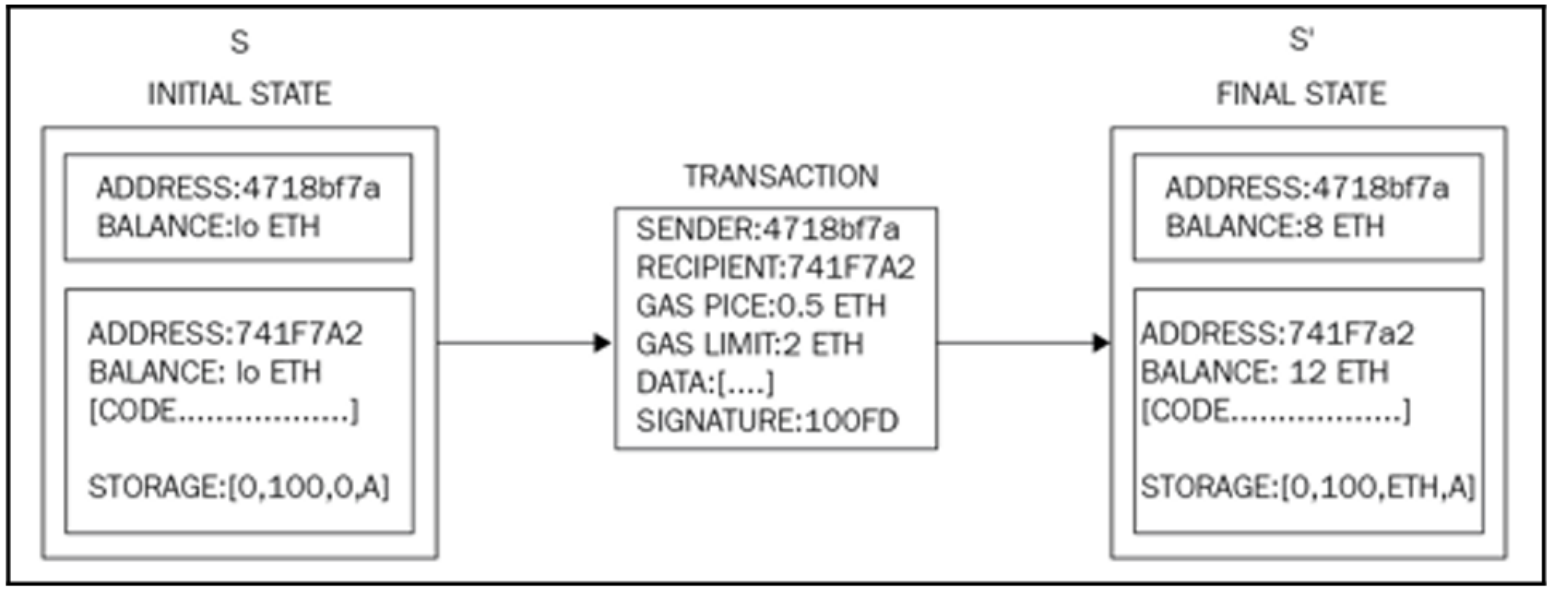
```
> var address =
"0xc7caf784fae5840bdc893b03b7391fce6efb6190"

> var myContract = eth.contract(abi).at(address)
```

Source: https://dzone.com/refcardz/getting-started-with-etherium-private-blockchain?chapter=1/

Ethereum Blockchain

Ethereum, just like any other blockchain, can be visualized as a transaction-based state machine. This definition is mentioned in the Ethereum yellow paper written by Dr. Gavin Wood.

The core idea is that in Ethereum blockchain, a genesis state is transformed into a final state by executing transactions incrementally. The final transformation is then accepted as the absolute undisputed version of the state. In the following diagram, the Ethereum state transition function is shown, where a transaction execution has resulted in a state transition:

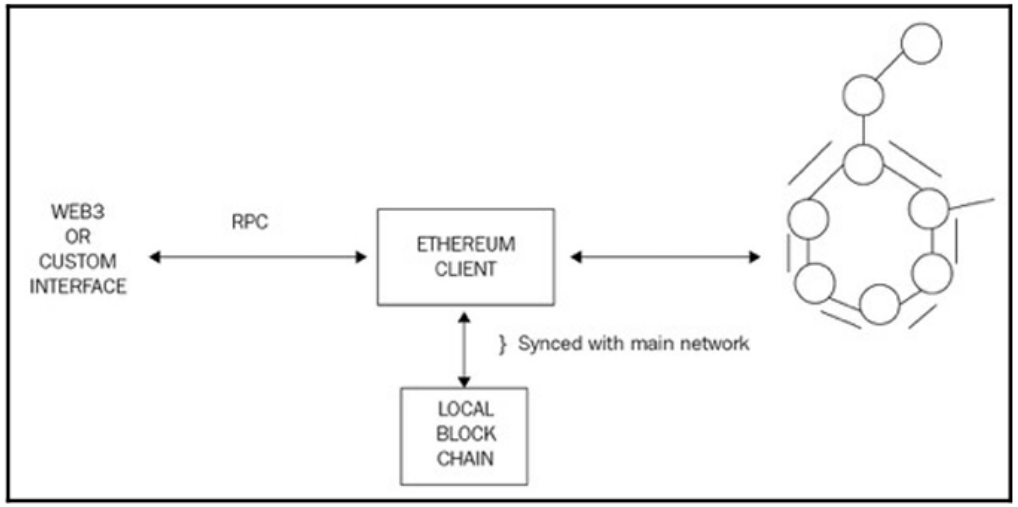


Source: Mastering Blockchain by Imran Bashir (Published by Packt.)

Ethereum Architecture

The Ethereum blockchain stack consists of various components. At the core, there is the Ethereum blockchain running on the peer-to-peer Ethereum network. Secondly, there's an Ethereum client (usually Geth) that runs on the nodes and connects to the peer-to-peer Ethereum network from where blockchain is downloaded and stored locally. It provides various functions, such as mining and account management. The local copy of the blockchain is synchronized regularly with the network. Another component is the `web3.js` library that allows interaction with the `geth` client via the **Remote Procedure Call** (**RPC**) interface.

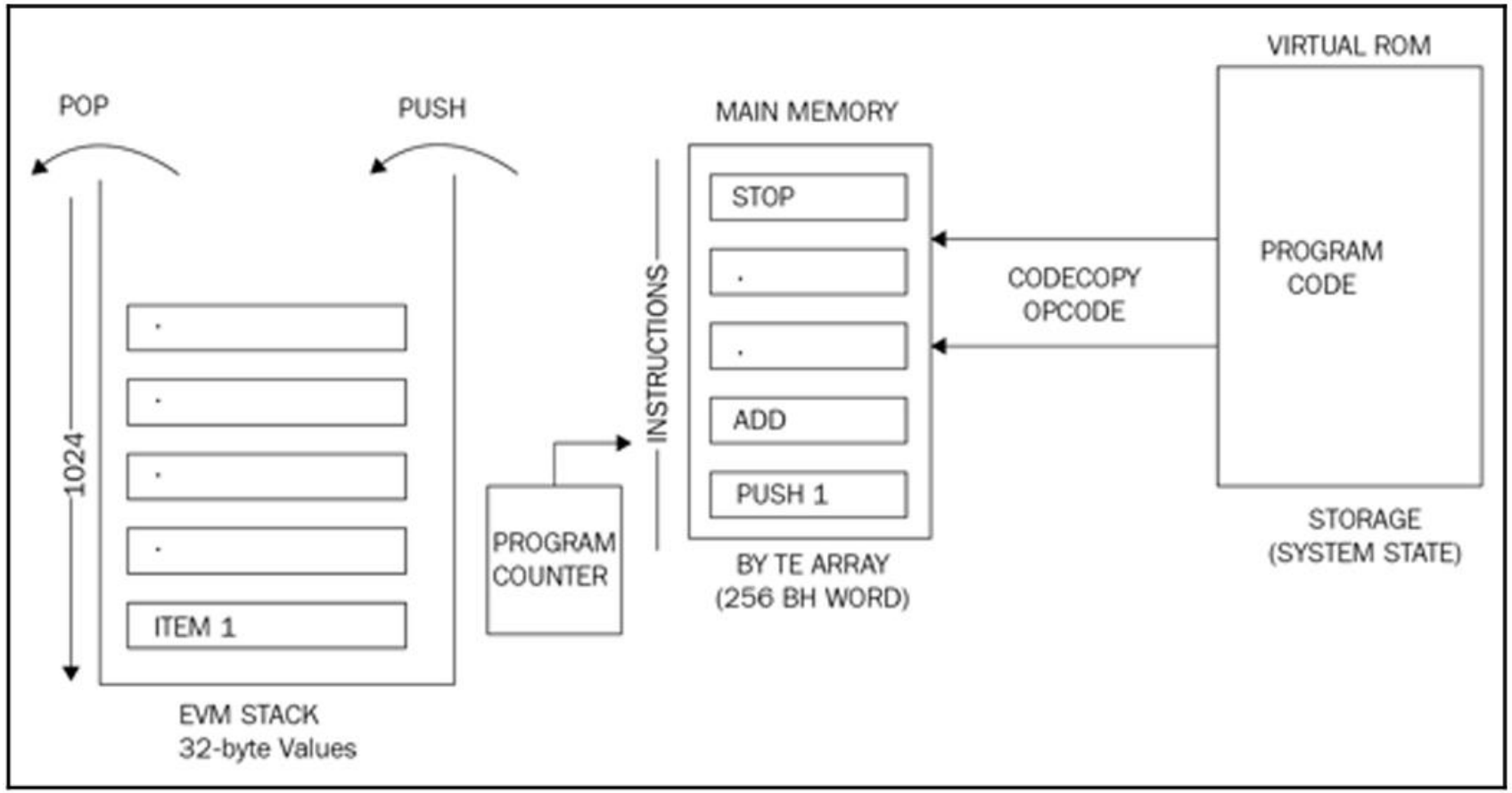
This architecture can be visualized in the following diagram:



The Ethereum stack showing various components

Source: Mastering Blockchain by Imran Bashir (Published by Packt.)

Ethereum EVM



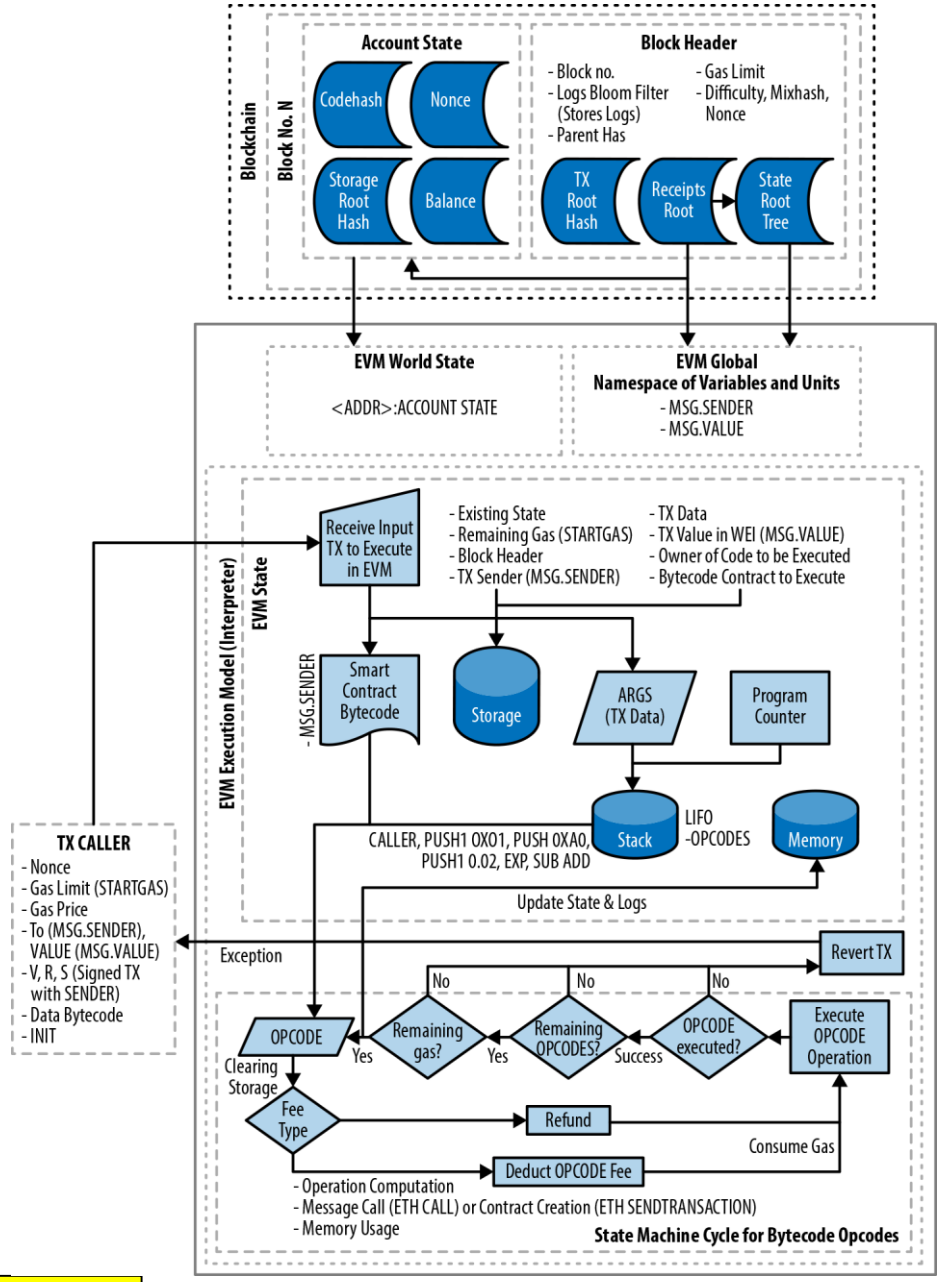
EVM operation

Source: Mastering Blockchain by Imran Bashir (Published by Packt.)

Ethereum EVM

This is the Ethereum Virtual Machine.

The EVM is also known as “The World Computer”



Source: Mastering Ethereum by Andreas Antonopolous & Gavin Wood

Ethereum DApp Architecture

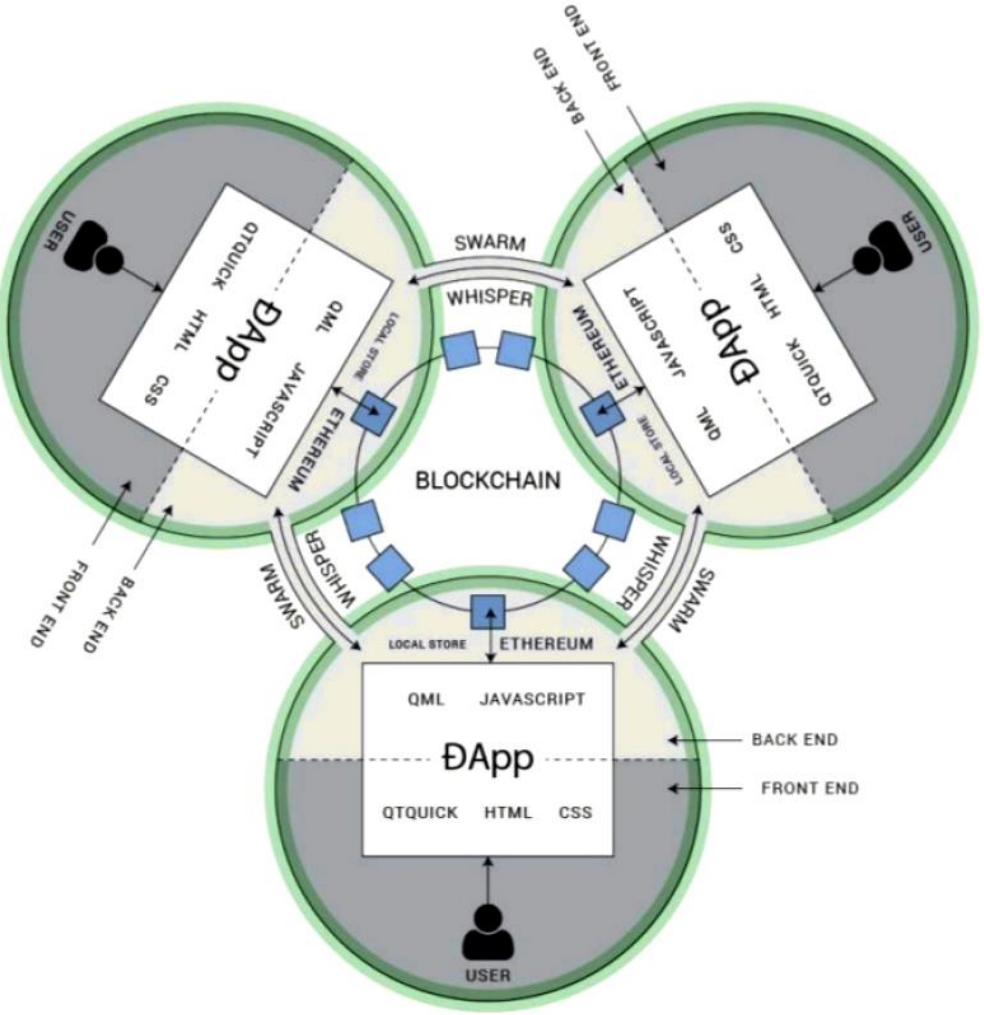


Fig. 11. Ethereum Architecture [52]

Source: https://www.researchgate.net/publication/315619465_A_more_pragmatic_Web_30_Linked_Blockchain_Data

Ethereum Web3.js Tech Stack

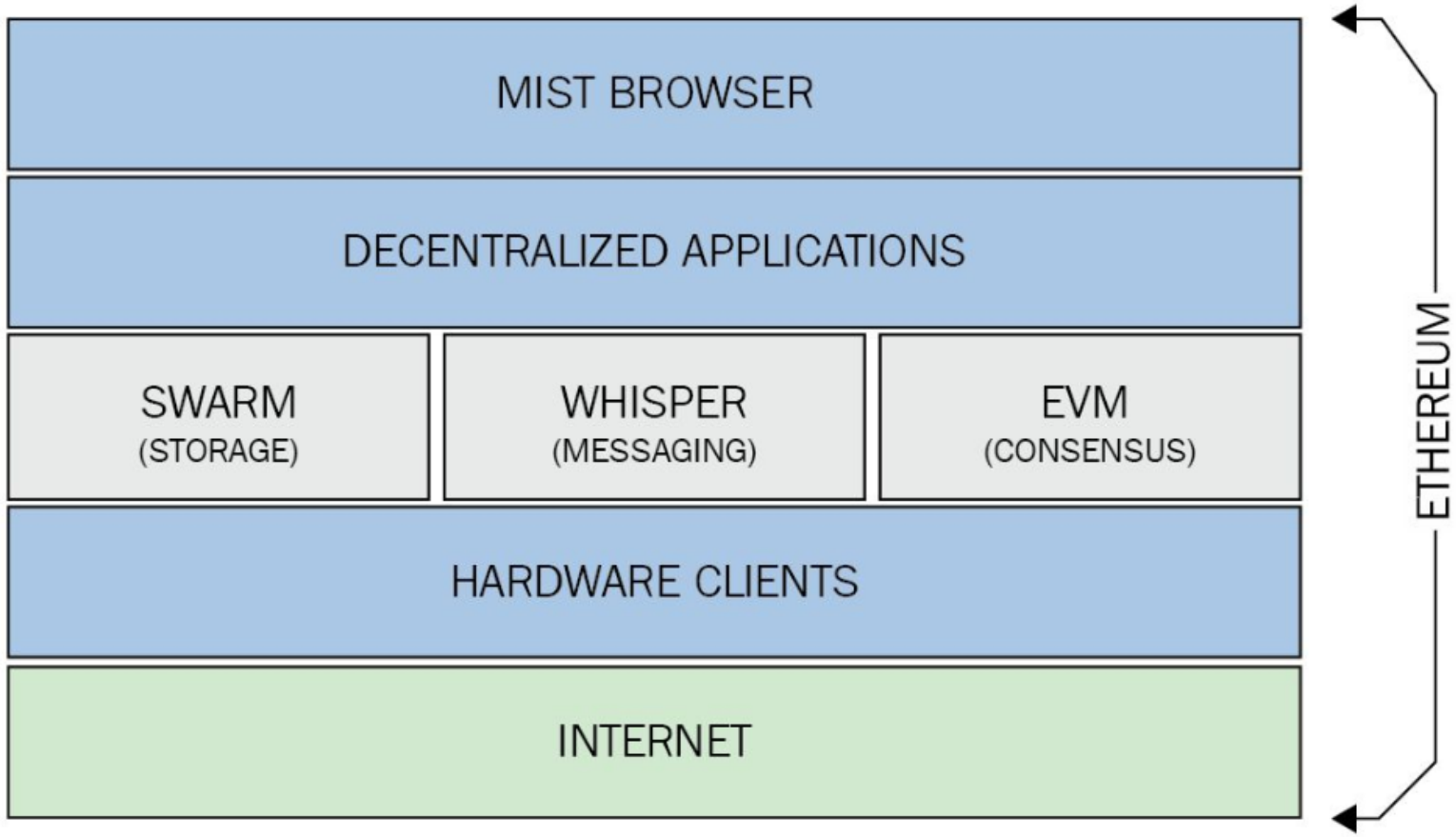
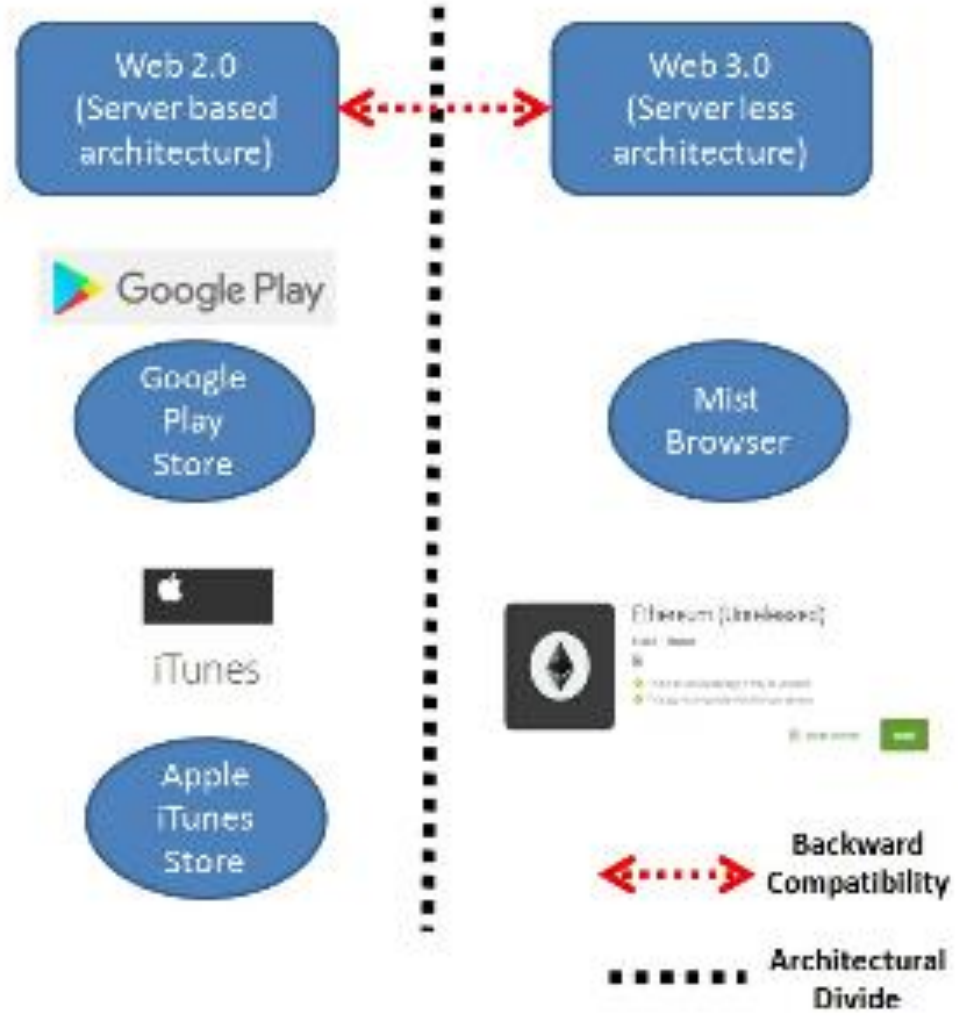


Figure 2.4: Web 3.0 tech stack for Ethereum, Source: Ethereum stack exchange

Web 2.0 Apps vs. Web3 DApps compared

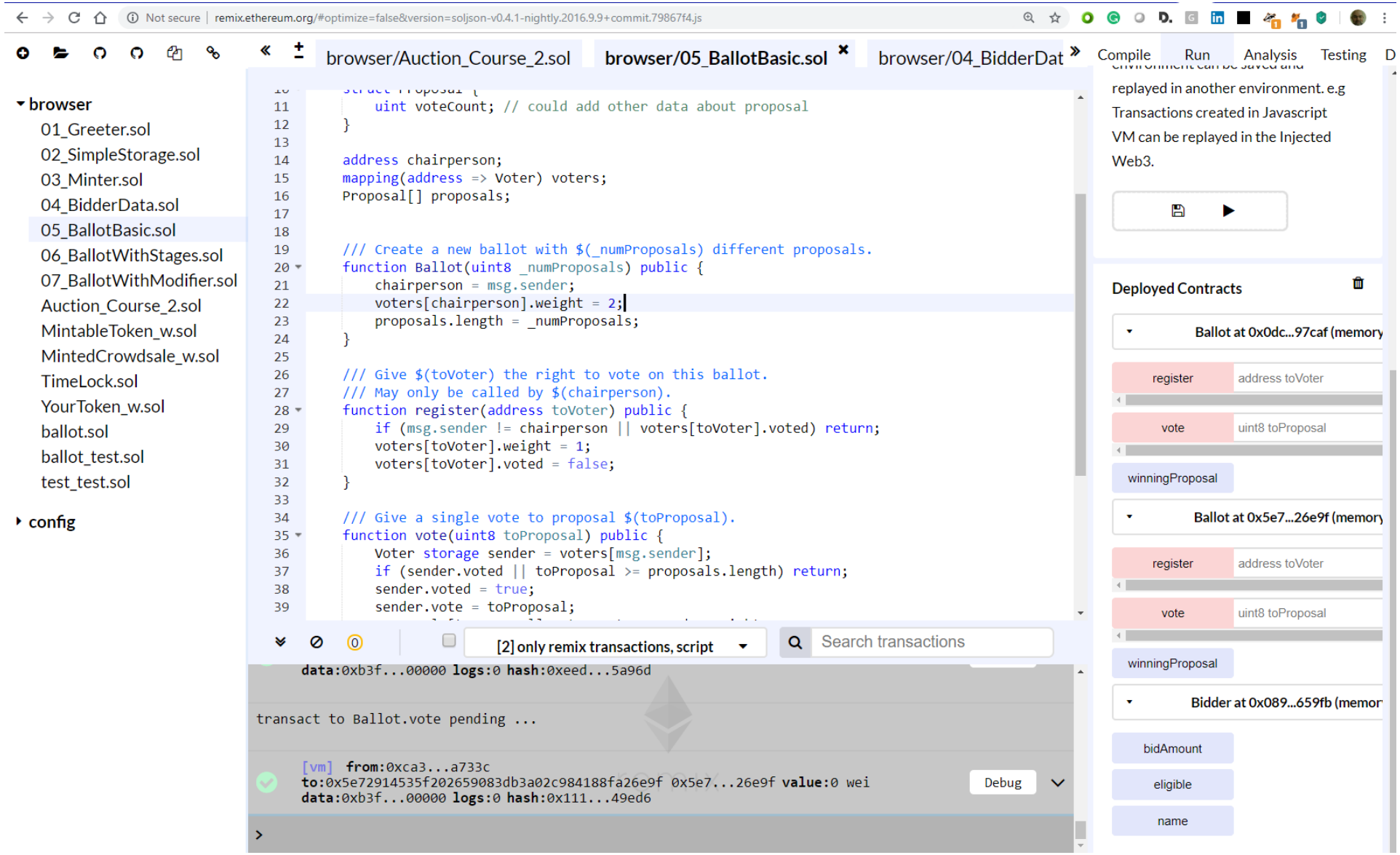


Ethereum DApp Development Languages

- **Solidity**
  - Solidity is a programming language that is intended for writing smart contracts for Ethereum-based blockchains. Solidity's syntax was based on JavaScript, which makes the language easier to pick up, and it also borrows concepts from C++ and Python.
  - Solidity is Function-based and uses Modifiers to control execution and Events to record what happens during the execution of a Solidity DApp.
- **Javascript**
- **Vyper (or Serpent)**
  - Like Python
- **LLL**
  - Lisp-Like Language

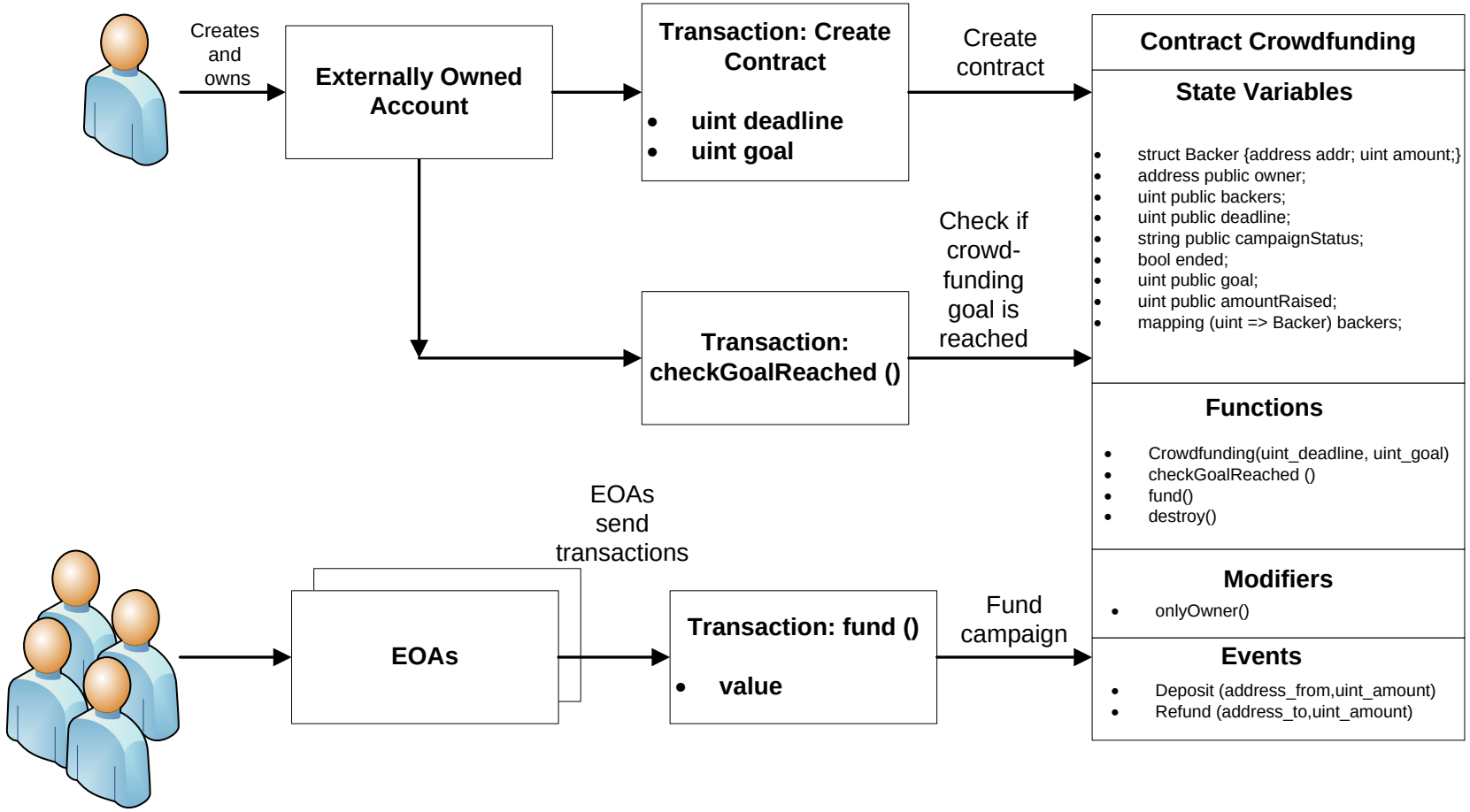
Special Note: All production EVM programs use “Gas” which is the measure of the cost of executing the DApp. This discourages misuse of EVM resources as well as sloppy or ineffective or inefficient or evil programming practices. Gas is measured in units of “Eth” and the smallest unit is a “Wei”, which is 10^{-18} Ether. The best Ethereum DApp Developers will thoroughly test their programs in advance and know what to expect in terms of the resource usage and consumption. It is considered sloppy programming to allow a program to run out of “Gas” before it has completed its designed mission.

Ethereum DApp Development – Solidity with the Remix Compiler



Example High-Level Implementation Diagram for a Solidity DApp

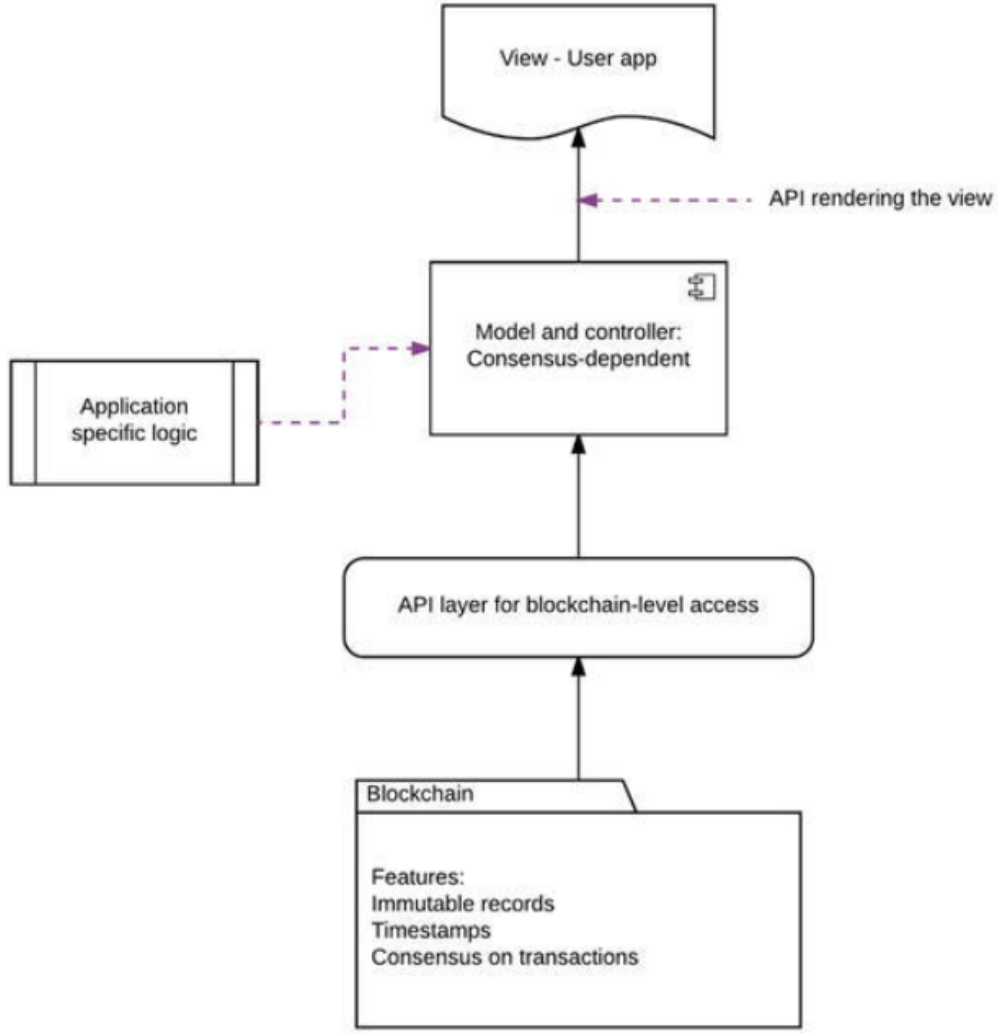
**(Example Business Case:
Crowdfunding Application)**



Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti

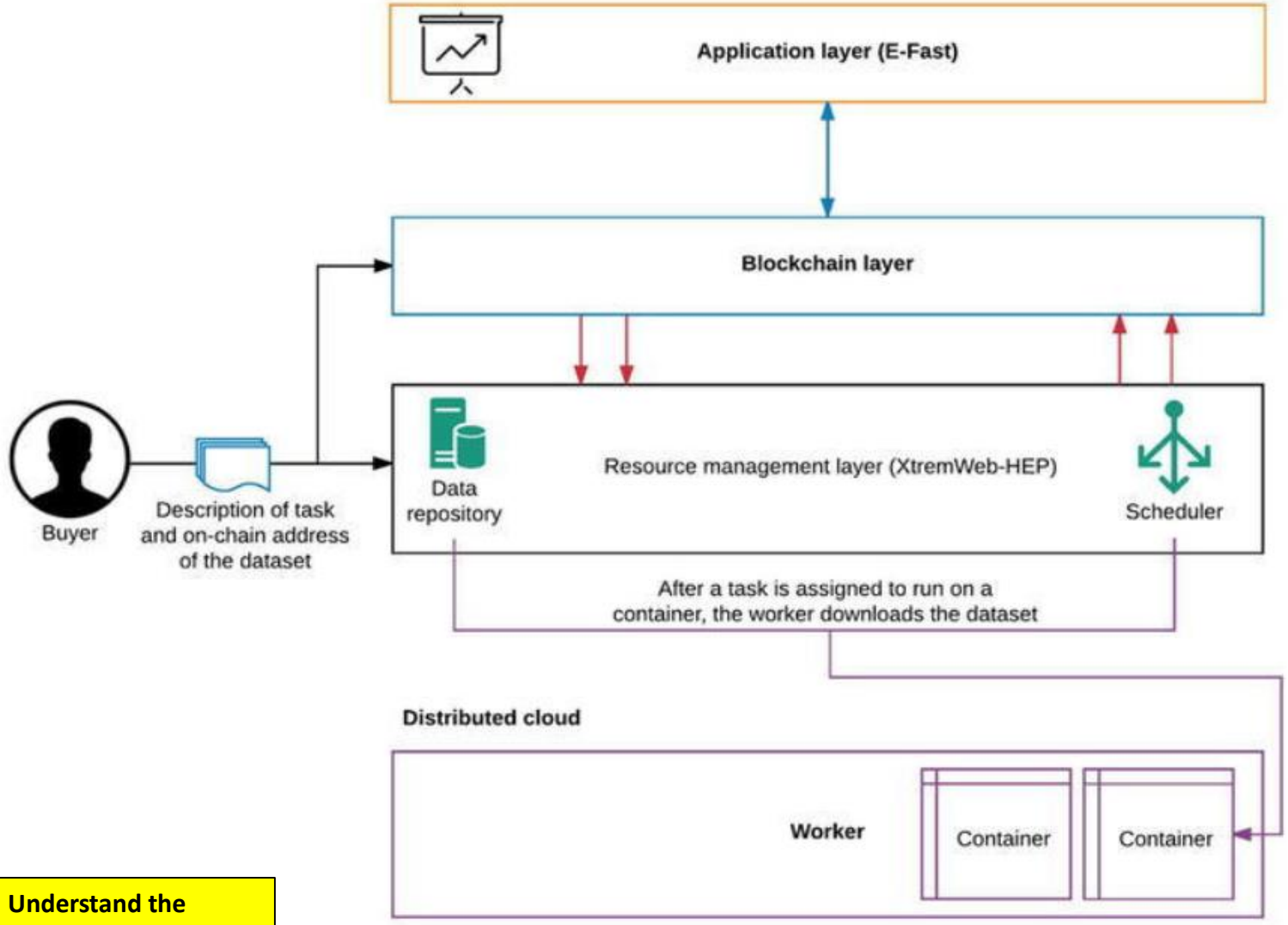
Topic 11: How to Design and Implement a Blockchain Solution Project – an Organized High-Level Step-by-Step Approach

Simple Blockchain Application Model



Source: **Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You** by Vikram Dhillon, David Metcalf, Max Hooper

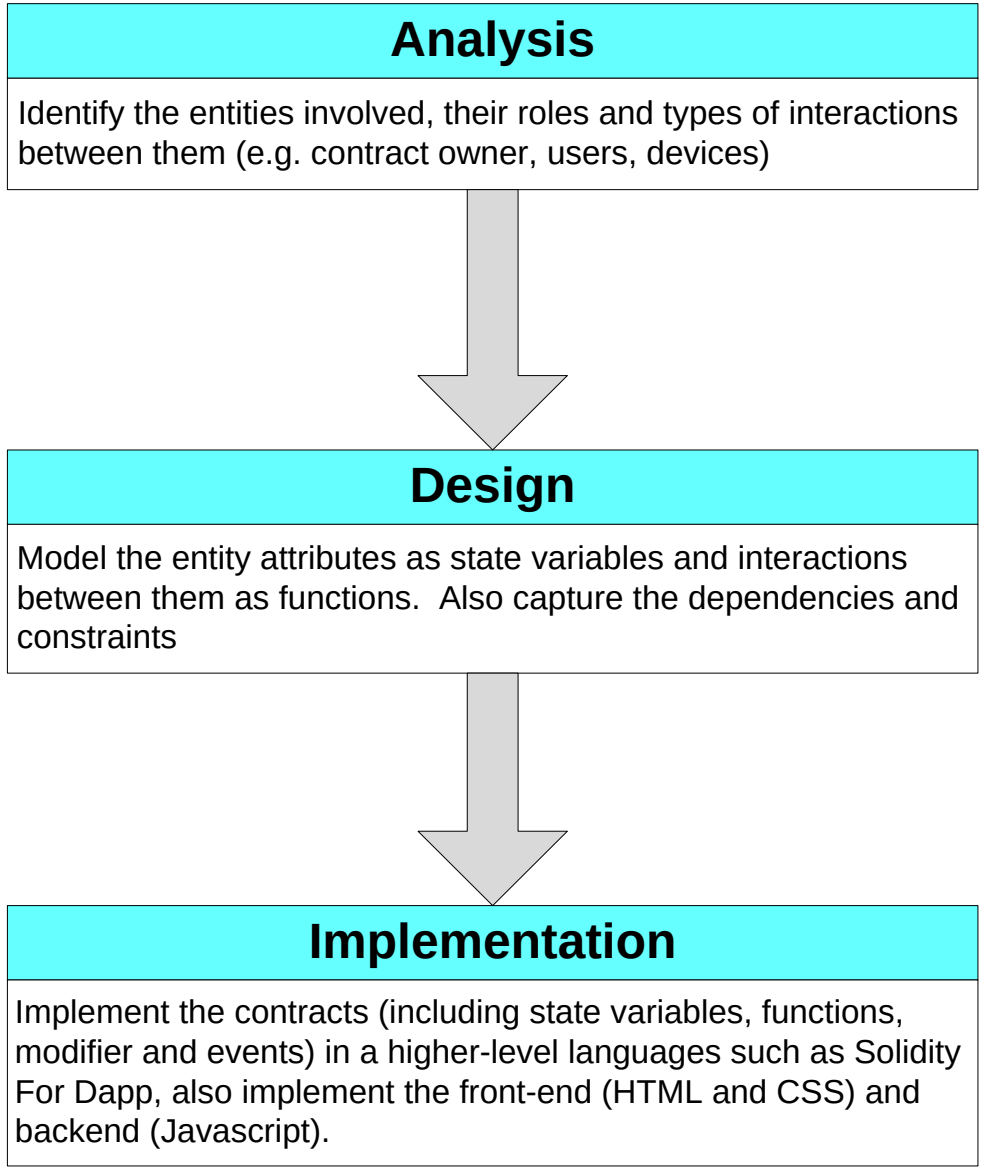
Example of a Blockchain Application



Source: **Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You** by Vikram Dhillon, David Metcalf, Max Hooper

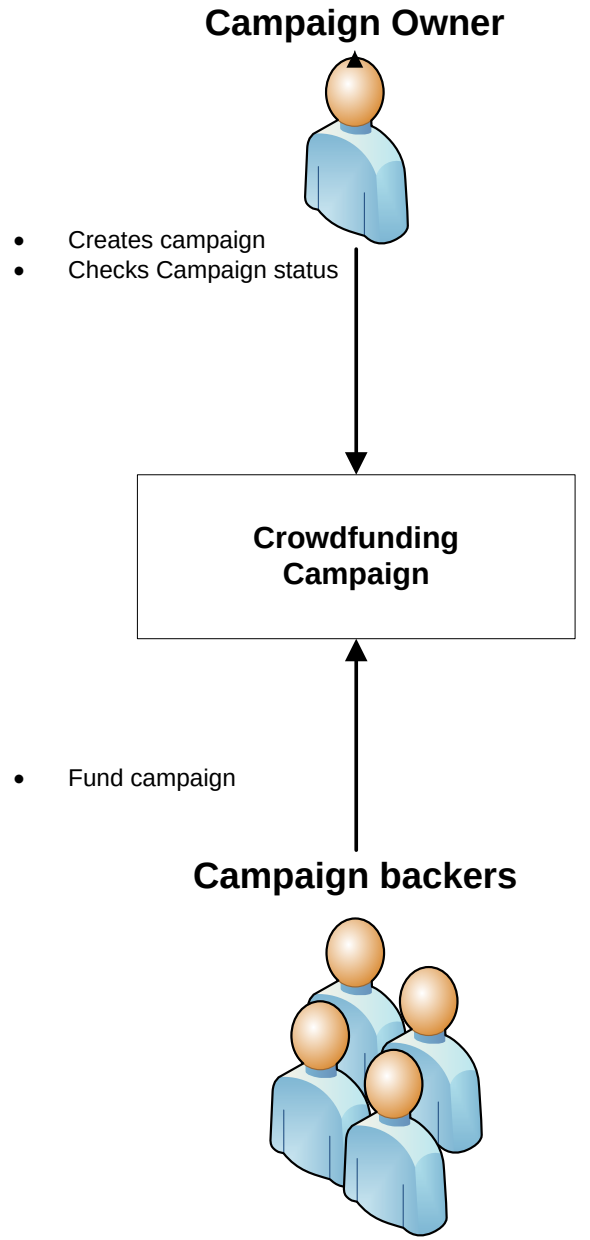
Creating a DApp

1. Analysis
2. Design
3. Implementation



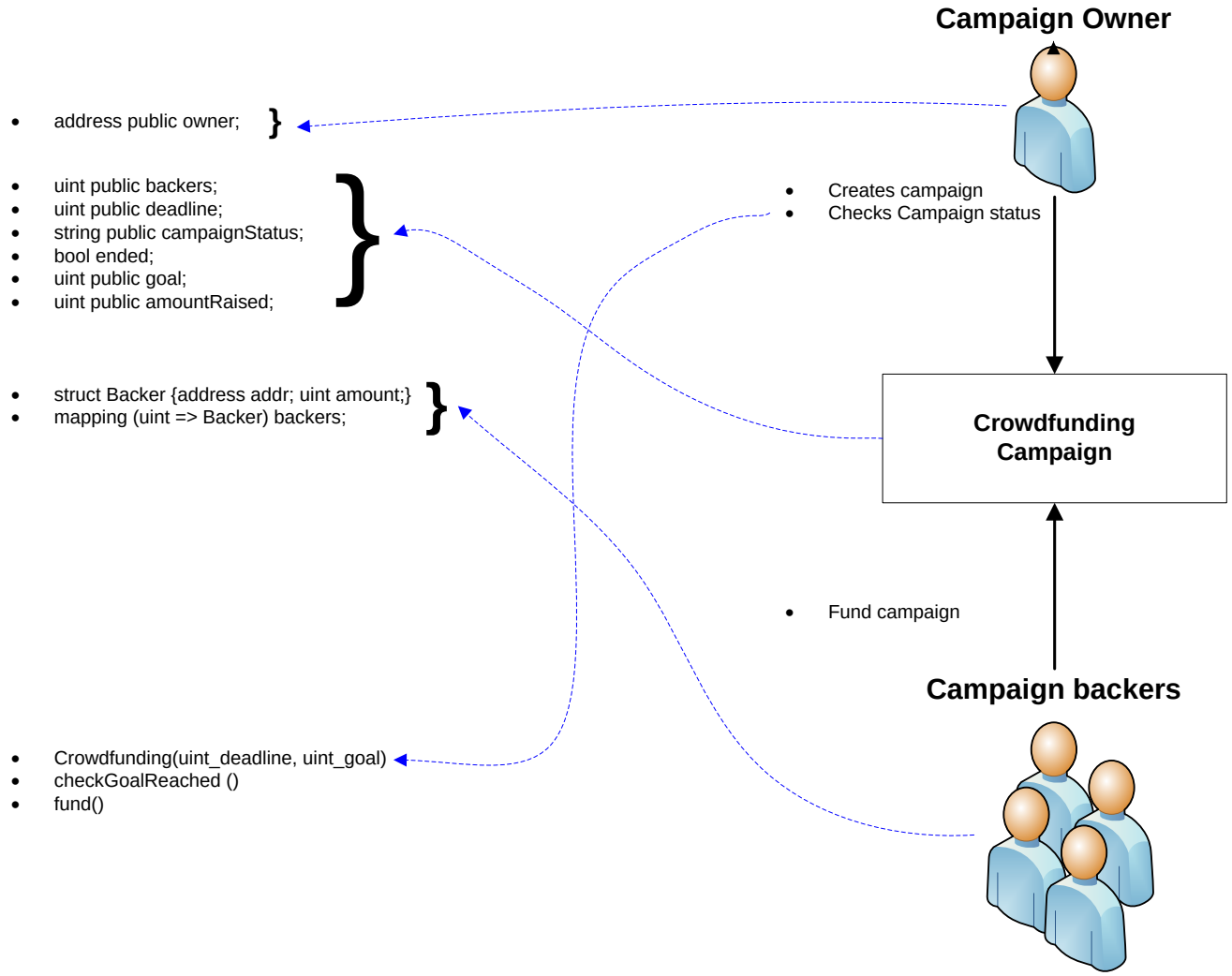
Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti

Creating a DApp - Analysis



Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti

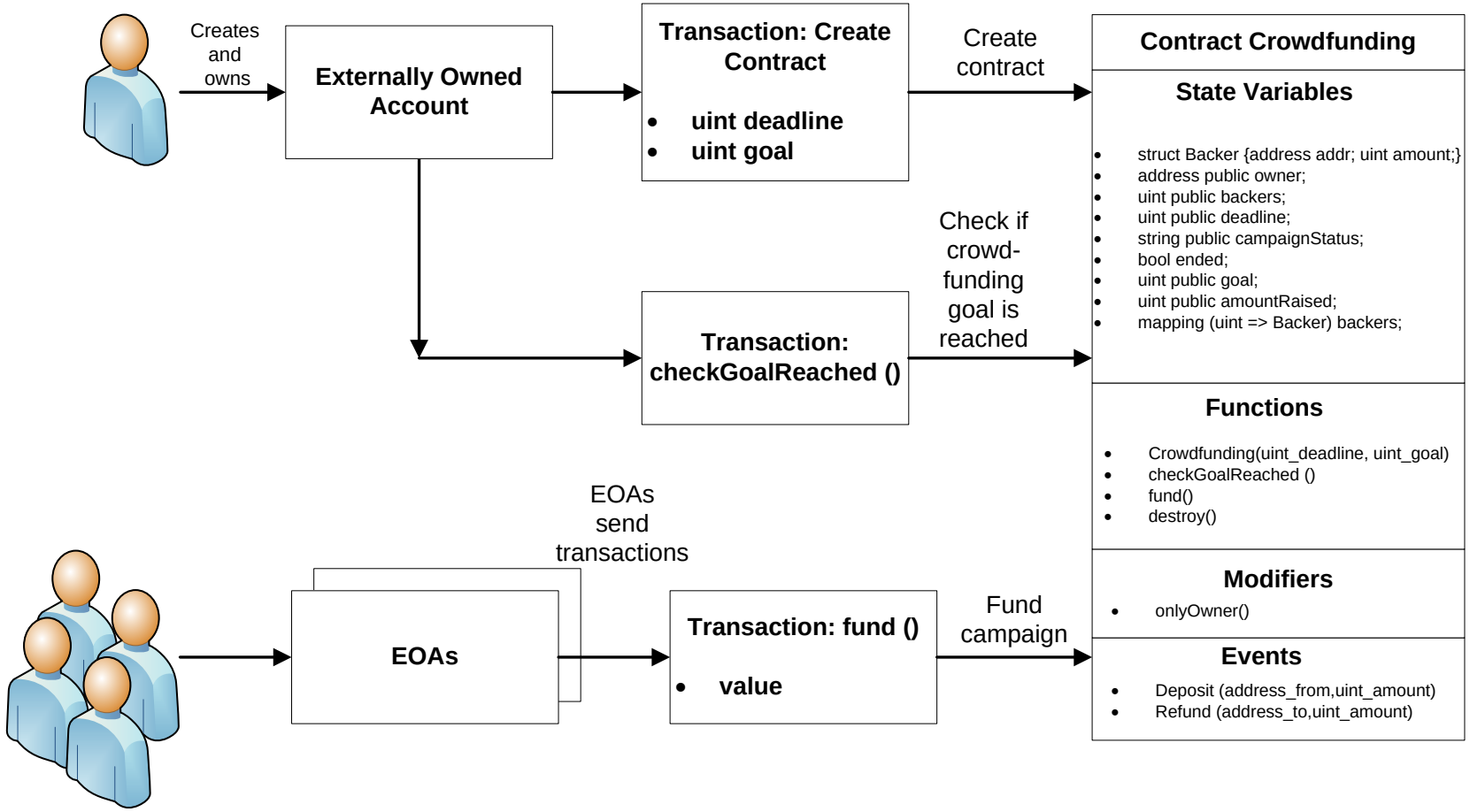
Creating a DApp - Design



Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti

Creating a DApp - High-Level Implementation Diagram

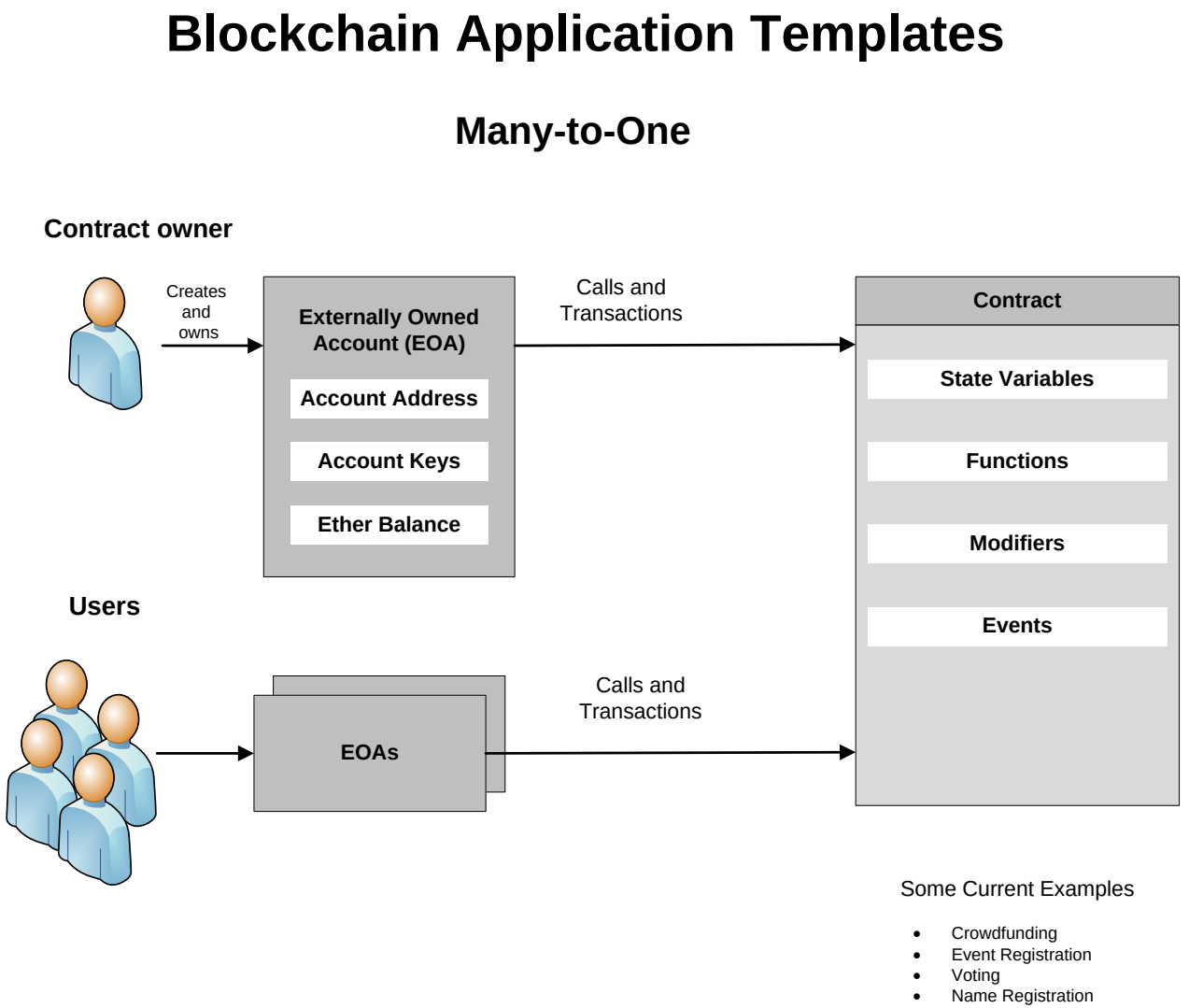
**(Example Business Case:
Crowdfunding Application)**



Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti

Best Practice – Using Templates and Patterns for Blockchain DApp Development

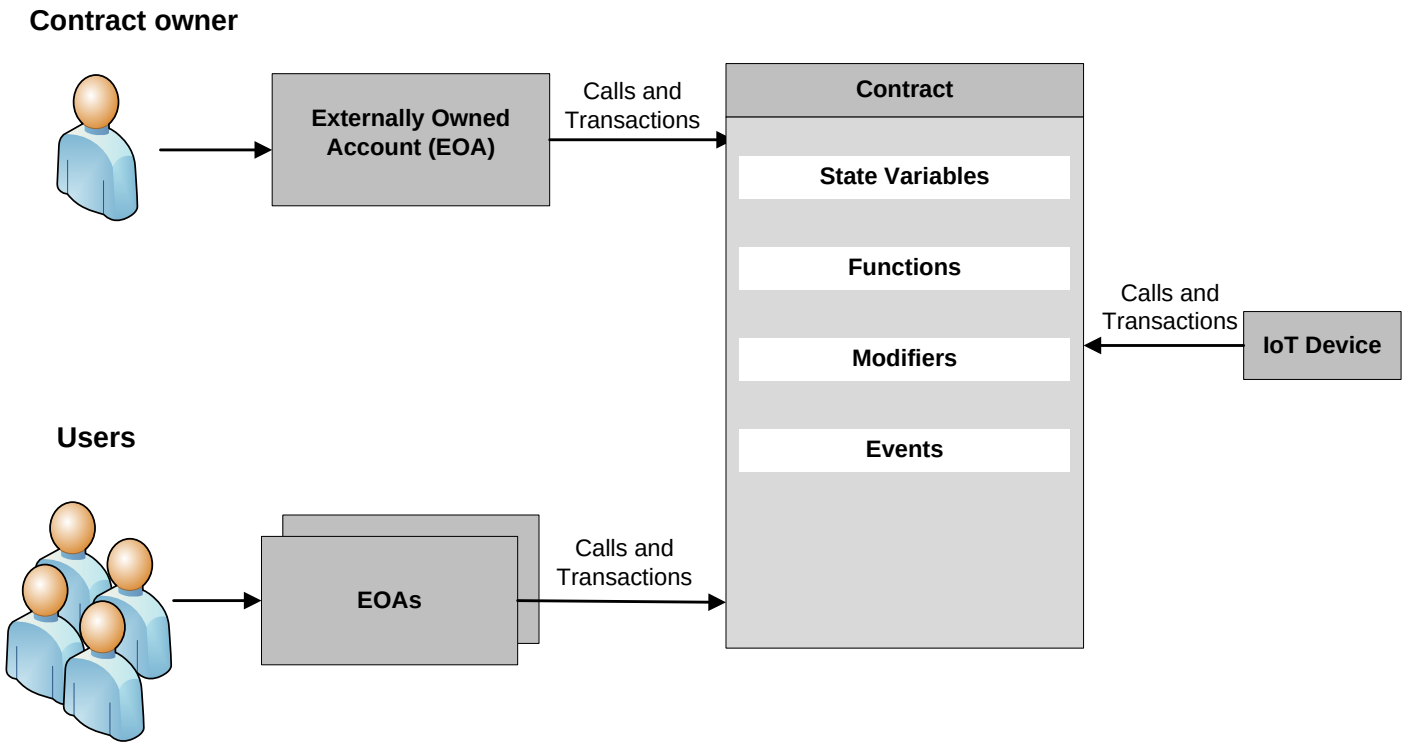
Blockchain Application Template – Many to One



Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti

Blockchain Application Template – Many to One for IoT Applications

Many-to-One for IoT Applications

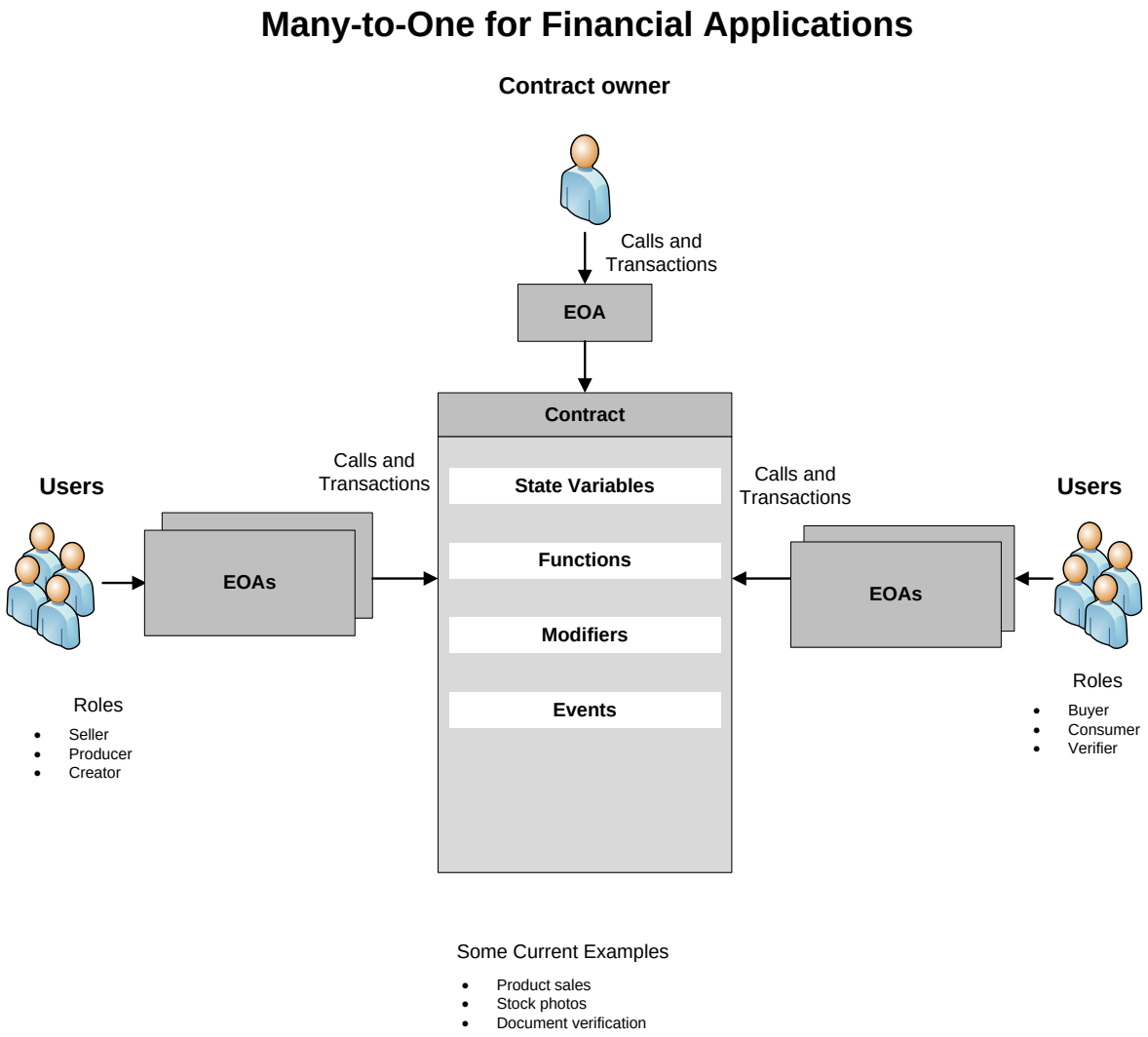


Some Current Examples

- Solar charging stations
- Smart switch

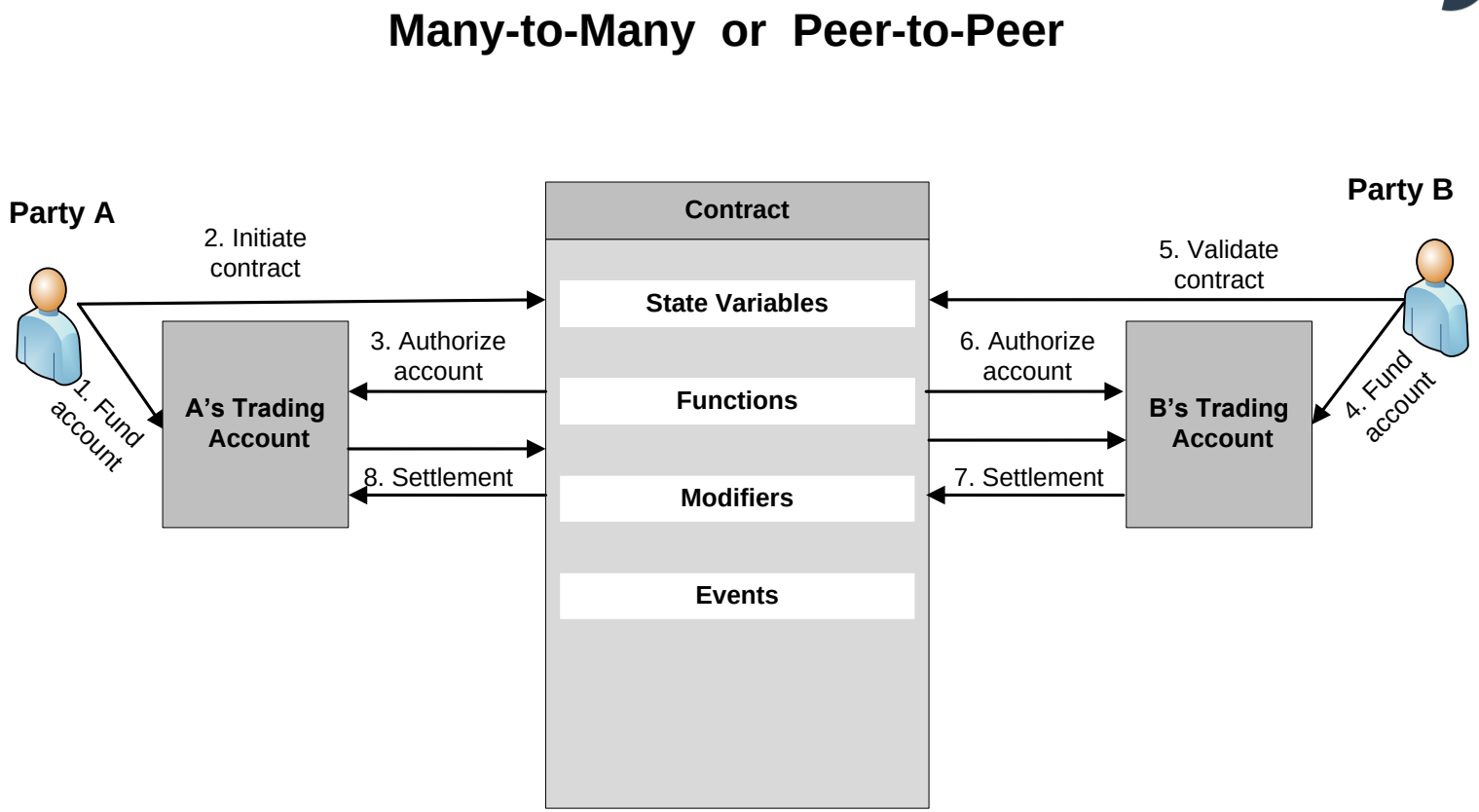
Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti

Blockchain Application Template – Many to One for Financial Applications



Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti

Blockchain Application Template – Many-to-Many or Peer-to-Peer



Some Current Examples

- Call option
- Interest rate swap

Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti

Common DApp Patterns

- Condition-Effects-Interaction
- Withdrawal
- Access Restriction
- Mortal
- Automatic Expiration
- Rejector
- Circuit Breaker
- Allow Once Per Account

Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti

Topic 12: How to Help your Organization Rapidly Ramp Up Skills and Readiness for Blockchain Application Development

The Required Skills for a Blockchain Development Staff

Blockchain Developer Skill Set
Top 30 Co-occurring IT Skills

For the 6 months to 12 July 2018, Blockchain Developer job roles required the following IT skills in order of popularity. The figures indicate the absolute number co-occurrences and as a proportion of all permanent job ads featuring Blockchain Developer in the job title.

1	397 (100.00%)	Blockchain	15	111 (27.96%)	Smart Contracts
2	200 (50.38%)	Finance	16	107 (26.95%)	Solidity
3	184 (46.35%)	JavaScript	17	106 (26.70%)	Linux
4	168 (42.32%)	Node.js	18	104 (26.20%)	AngularJS
5	151 (38.04%)	Ethereum	19	101 (25.44%)	Docker
6	146 (36.78%)	Bitcoin	20	98 (24.69%)	Redis
7	142 (35.77%)	SQL	21	93 (23.43%)	MySQL
8	139 (35.01%)	Cryptocurrency	21	93 (23.43%)	Banking
9	134 (33.75%)	Java	22	92 (23.17%)	Amazon AWS
10	125 (31.49%)	NoSQL	23	88 (22.17%)	HTML
11	123 (30.98%)	Git (software)	24	85 (21.41%)	Telecoms
12	122 (30.73%)	React	24	85 (21.41%)	PostgreSQL
13	118 (29.72%)	Test Automation	25	84 (21.16%)	Agile Software Development
13	118 (29.72%)	GitHub	25	84 (21.16%)	ES6
14	115 (28.97%)	Front End Development	26	77 (19.40%)	CSS

Additional Skills Required for a Blockchain Development Staff

- Web3.js
- DApp development
- UI and UX Design and Testing Skills
- Deep understanding of compiled code, Gas, and the Ethereum Virtual Machine (EVM)
- Secure coding
- Defensive coding
- Egoless Programming
- Stringent Code Reviews
- Networking
- Understanding of Protocols
- Planning
- Requirements
- Technical Specifications and Writing
- Design
- Architecture – Infrastructure, Data, and Security
- Testing – Testing – Testing
- Simulation
- Troubleshooting

And don’t forget
PROJECT MANAGEMENT &
PROGRAM MANAGEMENT!

The Challenges

- Huge Learning Curve
- DApps with Web3 and the EVM are not your Father's Web Developer Workbench
- You can really screw this up – easily
- Learning Egoless Programming
- Turnover – Once people get training and experience they may leave

Solving the Challenges & Winning

Find and utilize quality resources to accelerate your learning curve and immersion into the Blockchain World

Establish a Blockchain Expert or Champion imbued with the responsibility to be the Blockchain Evangelist

Build strong Learning Teams – Use Peter Senge's Learning Team Disciplines

Shared Vision

Personal Mastery

Mental Modeling

Team Learning

Systems Thinking

Stay abreast of Blockchain Technologies and Blockchain Politics and Blockchain Evolution

Join and participate in Local Blockchain Meetups

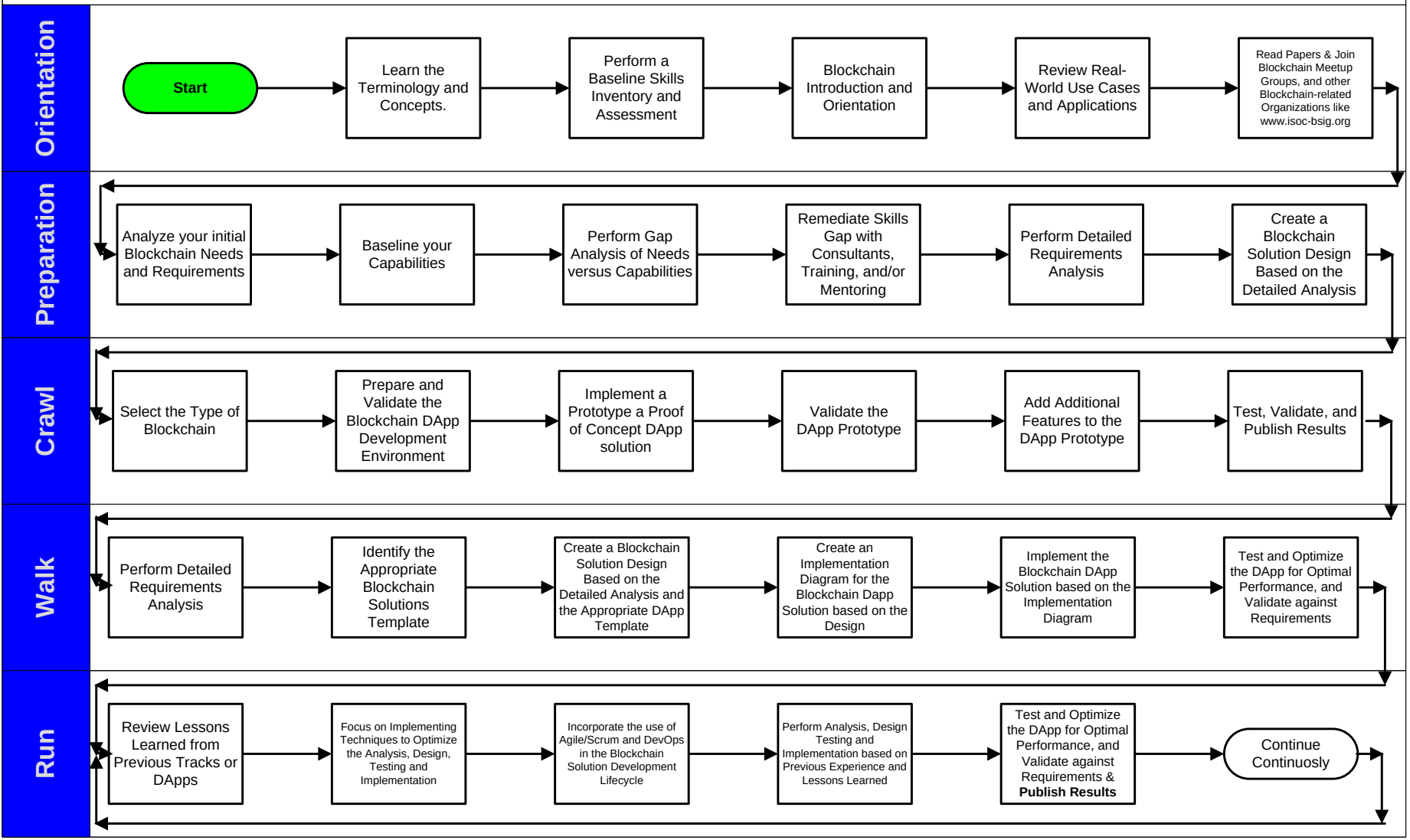
Go International - Get involved with the Internet Society and the Blockchain Special Interest Group - Both are free and the Blockchain SIG has great people and projects and leadership

www.internetsociety.com

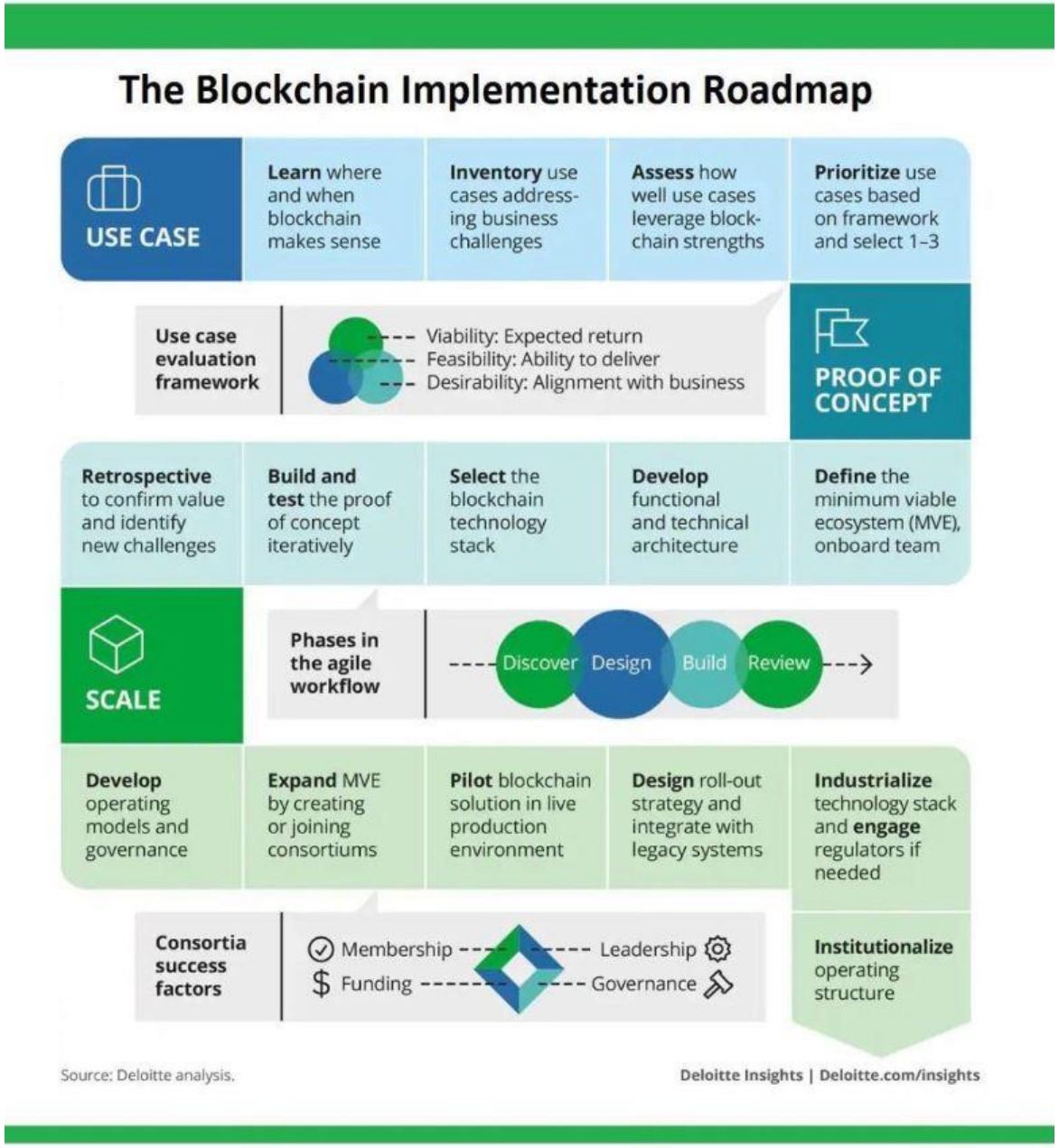
https://www.isoc-bsig.org/

https://www.linkedin.com/company/isoc-blockchain-sig/

Roadmap to "Blockchain" Your IT Organization: How to Help Your IT Staff Go from Square One to Competence & Dominance in Blockchain Technologies



Blockchain Implementation Roadmap



Conclusion – Day 1

Conclusion

We covered:

- History of Money and Conventional Ledger Functions
- Bitcoin Basics
- Tokenized Economy and Crypto Currency Concepts
- Blockchain Technology
- Ethereum Blockchain Technology
- Blockchain Beyond Bitcoin
- Blockchain Limits and Challenges
- Blockchain Security
- Examples of Real-world Blockchain Applications
- The Ethereum EVM, Smart Contracts, and Solidity
- How to Design and Implement a Blockchain Solution Project – an Organized High-Level Step-by-Step Approach
- How to Help your Organization Rapidly Ramp Up Skills and Readiness for Blockchain Application Development

Conclusion

From James Nguyen
February 12, 2019

Trust and Transparency

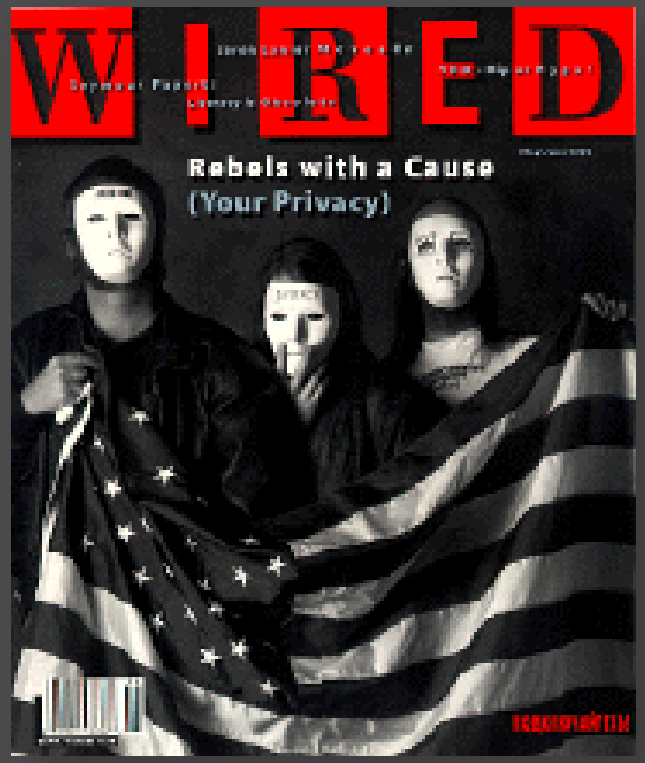
The bottom line is that it's not enough to just trust in blockchain security because there is usually more transparency than other technological data security and privacy methods. Developers, miners and even enterprises need to look at the entire digital ecosystem when considering security, as every single point provides savvy hackers with a weak leak to exploit.

As blockchain investment continues to skyrocket and the crypto markets continue to diversify — even with the recent slowdown — we will see more unique and sophisticated examples of cyber criminals penetrating blockchain's security veneer. That's the paradoxical ratio of technology: for as many positive innovations that tech brings up, there almost is an equal amount of sinister efforts to match it. The trick is to keep discussing the threats to blockchain while also inspiring and enabling the community to secure it.

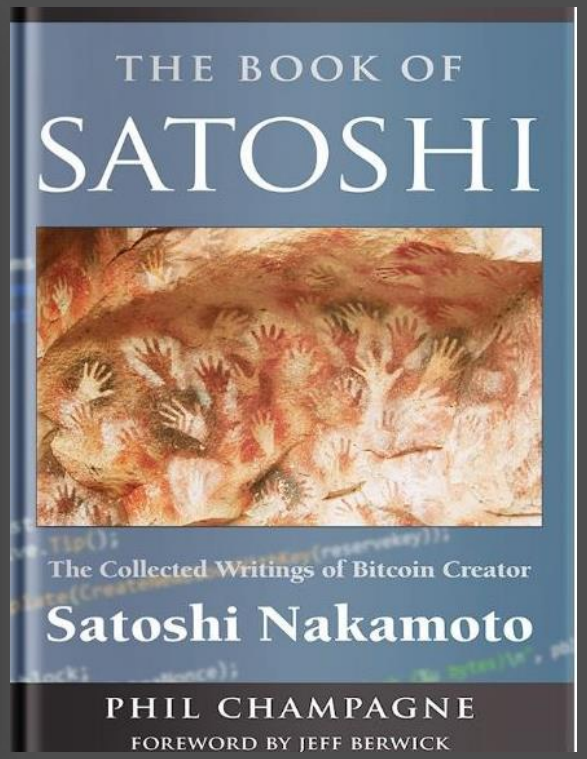
Source: **Blockchain still vulnerable to hacks despite security hype, but here are some solutions by James Nguyen. Retrieved from** **https://e27.co/blockchain-still-vulnerable-to-hacks-despite-security-hype-but-here-are-some-solutions-20190212/** -

Questions?

Questions?



Crypto Rebels
Revealed
Wired Magazine,
February 1993



Book of Satoshi
Collected Writings
Of Satoshi Nakamoto

General George S. Patton

References

References

•Antonopoulos, A. M. (2018). Mastering Bitcoin: Programming the Open Blockchain, second edition. Sebastopol, CA: O'Reilly Media, Inc.

•Antonopoulos, A. M. and Wood, G. (2019). Mastering Ethereum: Building Smart Contract sand DApps. Sebastopol, CA: O'Reilly Media, Inc.

•Associated Press. (2014). Mt. Gox finds 200,000 missing bitcoins. Retrieved from http://money.msn.com/business-news/article.aspx?feed=AP&date=20140321&id=17454291 on March 21, 2014.

•Bahga, A. and Madisetti, V. (2017). Blockchain Applications: A Hands-On Approach. Published by Arshdeep Bahga and Vijay Madisetti. www.blockchain-book.com .

•Bambara, J. J. and Allen P. R. (2018). Blockchain: A Practical Guide to Developing Business, Law, and Technology Solutions. New York, NY: McGraw-Hill Education.

•Bashir, I. (2018). Mastering Blockchain, second edition. Birmingham, UK: Packt Publishing Ltd.

•BBC. (2014). Troubled MtGox Bitcoin boss emerges after shut down Retrieved from http://www.bbc.com/news/technology-26352442 on February 26, 2014.

•Bitcoin.org. (2014). Bitcoin.org FAQs.. Retrieved from https://bitcoin.org/en/faq on April 10, 2014.

•Bitcoin Scammers. (2014). Bit Coin Scammers. Retrieved from http://bitcoinscammers.com/ on April 9, 2014.

•Casey, M. J. and Vigna, P. (2018). The Truth Machine: The Blockchain Reference and the Future of Everything. New York, NY: St. Martin's Press.

•Caughey, M. (2013). Bitcoin Step by Step, second edition. Amazon Digital Services.

References

•Champagne, P. (2014). The Book of Satoshi: The Collected Writings of Bitcoin Creator Satoshi Nakamoto. Published by E53 Publishing, LLC.

•Dannen, C. (2017). Introducing Ethereum and Solidity: Foundations of Crytocurrency and Blockchain Programming for Beginners. New York, NY: Apress

•De Filippi, P. and Wright, A. (2018). Blockchain and the Law: the Rule of Code. Cambridge, MA: President and Fellows of Harvard College.

•De Havilland, P. (2018). Greedy, Prodigal, and Suicidal — Hosho to Save Smart Contracts From Three Deadly Sins. An article published at Bitsonline.com on September 3, 2018. Retrieved from https://bitsonline.com/greedy-prodigal-suicidal-hosho-smart-contracts/ on February 27, 2019.

•Dhillon, V., Metcalf, D., and Hooper, M. (2017). Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Nake It Work for You. New York, NY: Apress.

•Drescher, D. (2017). Blockchain Basics. Frankfort am Main, Germany: Apress.

•Eddison, L. (2017). Ethereum: A Deep Dive into Ethereum. Published by Leonard Eddison.

•Etwaru, R. (2017). Blockchain Trust Companies. Indianapolis, IN: Dog Ear Publishing.

•Ferry, T. (2019). To Blockchain or not to Blockchain. An article publsihed at Medium.com on June 8, 2018. Retrieved on January 13, 2019 from https://medium.com/causys/to-blockchain-or-not-to-blockchain-aed05bf08150 .

•Gerard, D. (2107), Attack of the 50 Foot Blockchain: Bitcoin, Blockchain, Ethereum, and Smart Contracts. Published by David Gerard. www.davidgerard.co.uk/blockchain .

•GreenBerg, A. (2019). A BlockchainBandit Is Guessing Private Keys and Scoring Millions, An article published on April 23, 2019 at Wired.com and retrieved from https://www.wired.com/story/blockchain-bandit-ethereum-weak-private-keys/ on April 23, 2019.

References

•Hornyak, T. (2014). 'Malleability' attacks not to blame for Mt. Gox's missing bitcoins, study says. Retrieved from http://www.pcworld.com/article/2114200/malleability-attacks-not-to-blame-for-mt-goxs-missing-bitcoins-study-says.html on March 27, 2014.

•Incencio, R. (2014). Ransomware and Bitcoin Theft Combine in BitCrypt. Retrieved from http://blog.trendmicro.com/trendlabs-security-intelligence/ransomware-and-bitcoin-theft-combine-in-bitcrypt/ on March 27, 2014.

•Laurence, T. (2017). Blockchain for Dummies. Hoboken, NJ: John Wiley & Sons, Inc.

•Lee, T. B. (2013). 12 questions about Bitcoin you were too embarrassed to ask. Retrieved from http://www.washingtonpost.com/blogs/the-switch/wp/2013/11/19/12-questions-you-were-too-embarrassed-to-ask-about-bitcoin/ on November 19, 2013.

•Ma, M. (2017). Blockchain Design Sprint: An Agile Innovation Workbook to Implement an Agile Design Sprint for your Blockchain Business. Published by Future Lab www.futurelabconsulting.com .

•Markowitz, E. (2014). Cryptocurrencies Are the New Spam Frontier. Retrieved from http://www.vocativ.com/tech/bitcoin/cryptocurrencies-new-spam-frontier/ on March 28, 2014.

•Nakamoto. S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf on November 1, 2013.

•Nguyen, J. (2019). Blockchain still vulnerable to hacks despite security hype, but here are some solutions. Retrieved from https://e27.co/blockchain-still-vulnerable-to-hacks-despite-security-hype-but-here-are-some-solutions-20190212/ on February 13, 2019.

•O'Ham, T. (2018). Singapore Research Team Codifies 3 new Ethereum VM Vulnerabilities. An article published at Bitsonline.com on February 21, 2018. Retrieved from https://bitsonline.com/singapore-research-ethereum/ on February 27, 2019.

•Orcutt, M. (2019). Once Hailed as Unhackable, Blockchains Are now Getting Hacked. An article in MIT Review. Published February 19, 2019. Retrieved from https://www.technologyreview.com/s/612974/once-hailed-as-unhackable-blockchains-are-now-getting-hacked/ on February 24, 2019.

References

•Popper, N. (2013). Into the Bitcoin Mines, Retrieved from http://dealbook.nytimes.com/2013/12/21/into-the-bitcoin-mines/?hp&_r=0 on December 21, 2013.

•Prusty, N. (2017). Building Blockchain Projects: Building Decentralized Blockchain Applications with Ethereum and Solidity. Birmingham, UK: Pact Publishing.

•Ramone, A. D. (2019). How to Secure a Blockchain: 3 Things Business Leaders Know. An article publisged at Techrepublic.com on April 18, 2019. Retrieved from https://www.techrepublic.com/article/how-to-secure-a-blockchain-3-things-business-leaders-need-to-know/ on April 23, 2019.

•SCGNEWS. (2014). The IRS Just Declared War on Bitcoin - Retroactively. Retrieved from http://scgnews.com/the-irs-just-declared-war-on-bitcoin-retroactively on March 27, 2014.

•Sharkey, T. (2014. Inside Bitcoins NYC Day 1: Bitcoin 2.0 Takes Center Stage. Retrieved from http://www.coindesk.com/inside-bitcoins-nyc-day-1-bitcoin-2-0-takes-center-stage/ on April 8, 2014.

•Zenko, M. (2017). Bitcoins for Bombs – a Blog published at the Council on Foreign Relations on August 17, 2017. Retrieved from https://www.cfr.org/blog/bitcoin-bombs on February 13, 2019.

References – Best Blockchain Books

- **Mastering Ethereum**

– by Andreas M. Antonopoulos and Dr. Gavin Wood.

- **Mastering Blockchain - Second Edition**

–by Imran Bashir

- **Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners**

–By Chris Dannen

- **Blockchain Applications: A Hands-On Approach**

–by Arshdeep Bahga and Vijay Madisetti

- **Ethereum, Tokens & Smart Contracts: Notes on getting started**

–by Eugenio Noyola

- **Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You**

–by Vikram Dhillon, David Metcalf, Max Hooper

- **Truffle Quick Start Guide**

–by Nikhil Bhaskar

- **Foundations of Blockchain**

- **By Koshik Raj**

- **The Book of Satoshi: The Collected Writings od Bitcoin Creator Satoshi Nakamoto**

–By Phil Champagne

References – 12 Free Blockchain Resources

1. William Slater's Blockchain Resource Page http://billslater.com/blockchain
2. Factom University http://www.factom.com/university
3. Ethereum 101 http://www.ethereum101.org
4. Build on Ripple http://ripple.com/build
5. Programmable money by Ripple https://goo.gl/g8vFPL
6. DigiKnow https://youtu.be/scr68zFddso
7. Blockchain University http://blockchainu.co
8. Bitcoin Core https://bitcoin.org
9. Blockchain Alliance http://www.blockchainalliance.org
10. Multichain Blog http://www.mutichain,com/blog
11. HiveMind http://bitcoinhivemind.com
12. Chicago Blockchain Project http://chicagoblockchainproject.com/
13. Chicago Bitcoin and Open Blockchain Meetup Group https://www.meetup.com/Bitcoin-Open-Blockchain-Community-Chicago/

References – 10 Rules to Never Break the Blockchain

1. Don’t use Cryptocurrency or Blockchain to Skirt the Law
2. Keep your contracts as simple as possible
3. Publish with great caution
4. Back Up, Back Up, Back Up Your Private Keys
5. Triple-check the Address Before Sending Currency
6. Take Care When Using Exchanges
7. Beware Wi-Fi
8. Identify Your Blockchain Dev
9. Don’t Get Suckered
10. Don’t Trade Tokens Unless You Know What You’re Doing

References – 10 Free Blockchain Projects

- The R3 Consortium http://www.r3cev.com
- T ZERO: Overstocking the Stock Market http://www.overstock.com
- Blockstream's Distributed Systems http://www.blockstream.com
- OpenBazaar's Blockchain http://www.openbazaar.com
- Code Valley: Find Your Coder http://www.codevalley.com
- Bitfury's Digital Assets http://www.bitfury.com
- Any Coin Can Shapeshift http://www.shapeshift.io
- Machine-Payable Apps on 21 http://www.21.co
- Anonymous Transactions on Dash http://www.dash.org
- ConsenSys: Decentralized Applications: http://www.consensys.net

Practical Exercises

Practical Exercise 01

Create a Hash

1. Visit this website and type information about yourself or a message, and use the SHA 256 hash algorithm to create a hash http://www.hashemall.com/
2. Save the hash value.
3. Visit this website to decrypt your hash message: http://md5decrypt.net/en/Sha256/

Practical Exercise 02

Decode a Hash

Hash: **9ec4c12949a4f31474f299058ce2b22a**

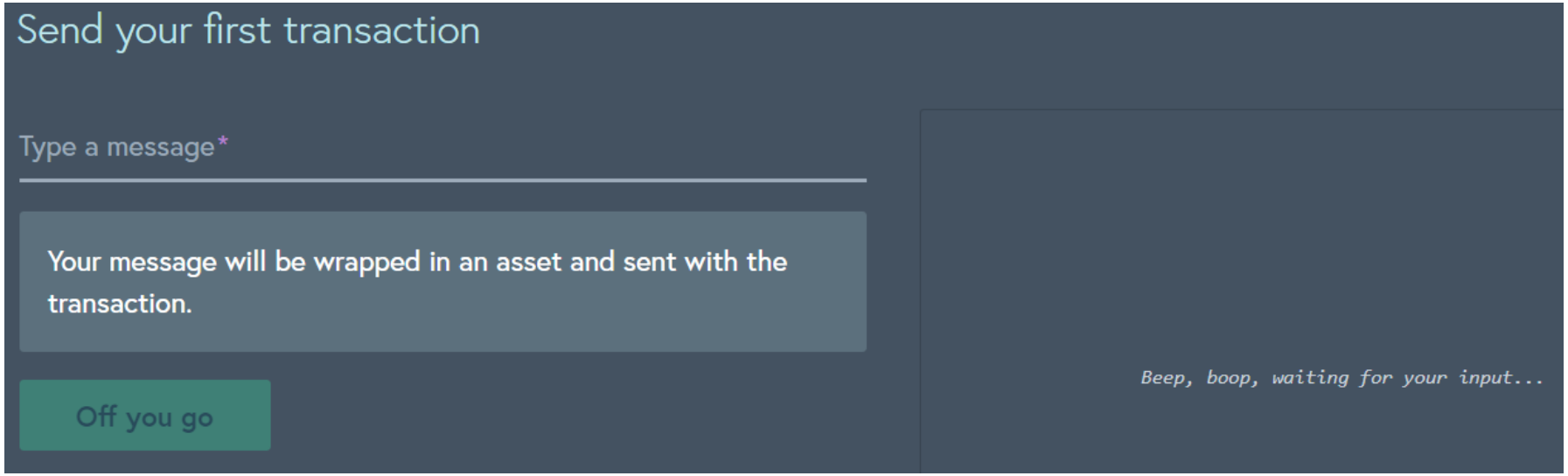
This hash is found on the emblem of U.S. Cybercommand. It is a message that was hashed

Using a commonly known hashing algorithm. Use this website to see if you can decrypt this Hash and see the message:
http://www.hashemall.com/

Practical Exercise 03

Create a Blockchain record using BigchainDB

- Visit this website and create your first Blockchain record:
- https://www.bigchaindb.com/getstarted/
- Copy and Save the results to a local text file named:
  YYYY_MMDD_FirstName_LastName_My_First_Blockchain_Transaction_.txt



Practical Exercise 04

Download and install Geth, the Ethereum Blockchain software

- Visit this website, to download Geth:
- https://geth.ethereum.org/downloads/

2. Install Geth into a directory you will create: c:\ethereum

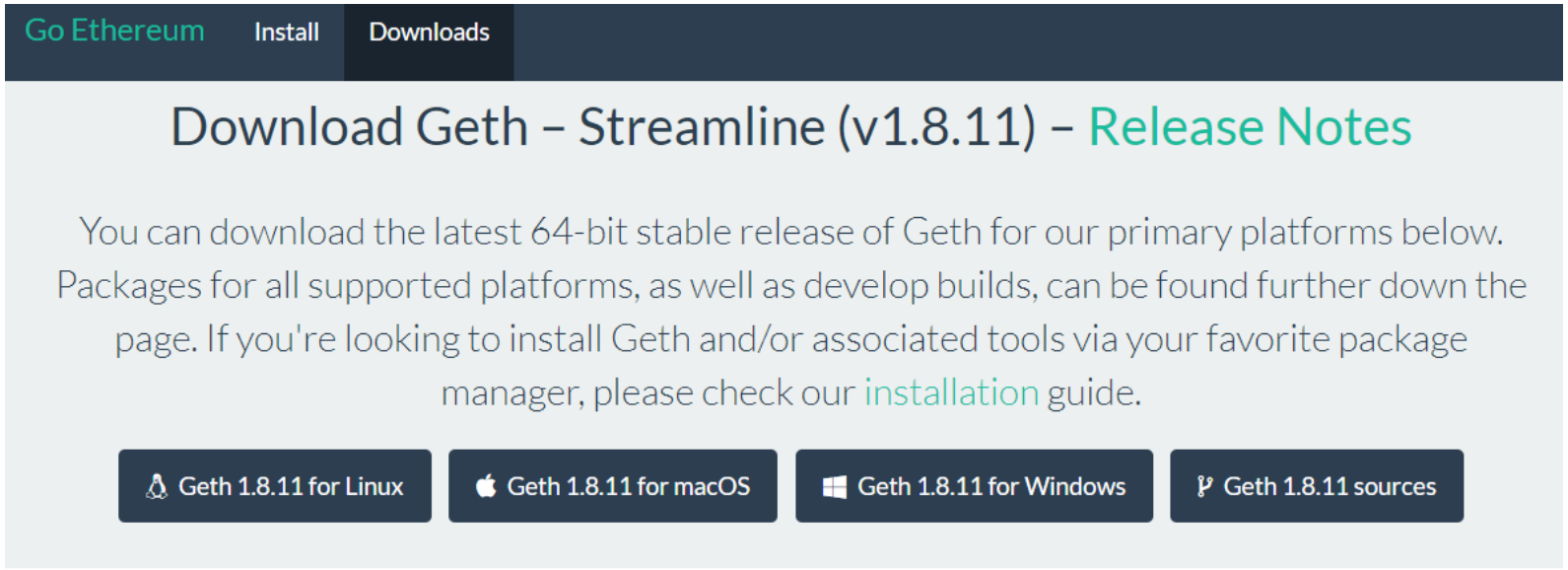
3. At the command line, launch Geth in testnet mode

4. Switch to miner mode

5. Extra Credit: if you set up an Ethereum Account, you can actually write data (like your name) to the Ethereum Blockchain and view it

Practical Exercise 04

Download Geth



Specific Versions

If you're looking for a specific release, operating system or architecture, below you will find:

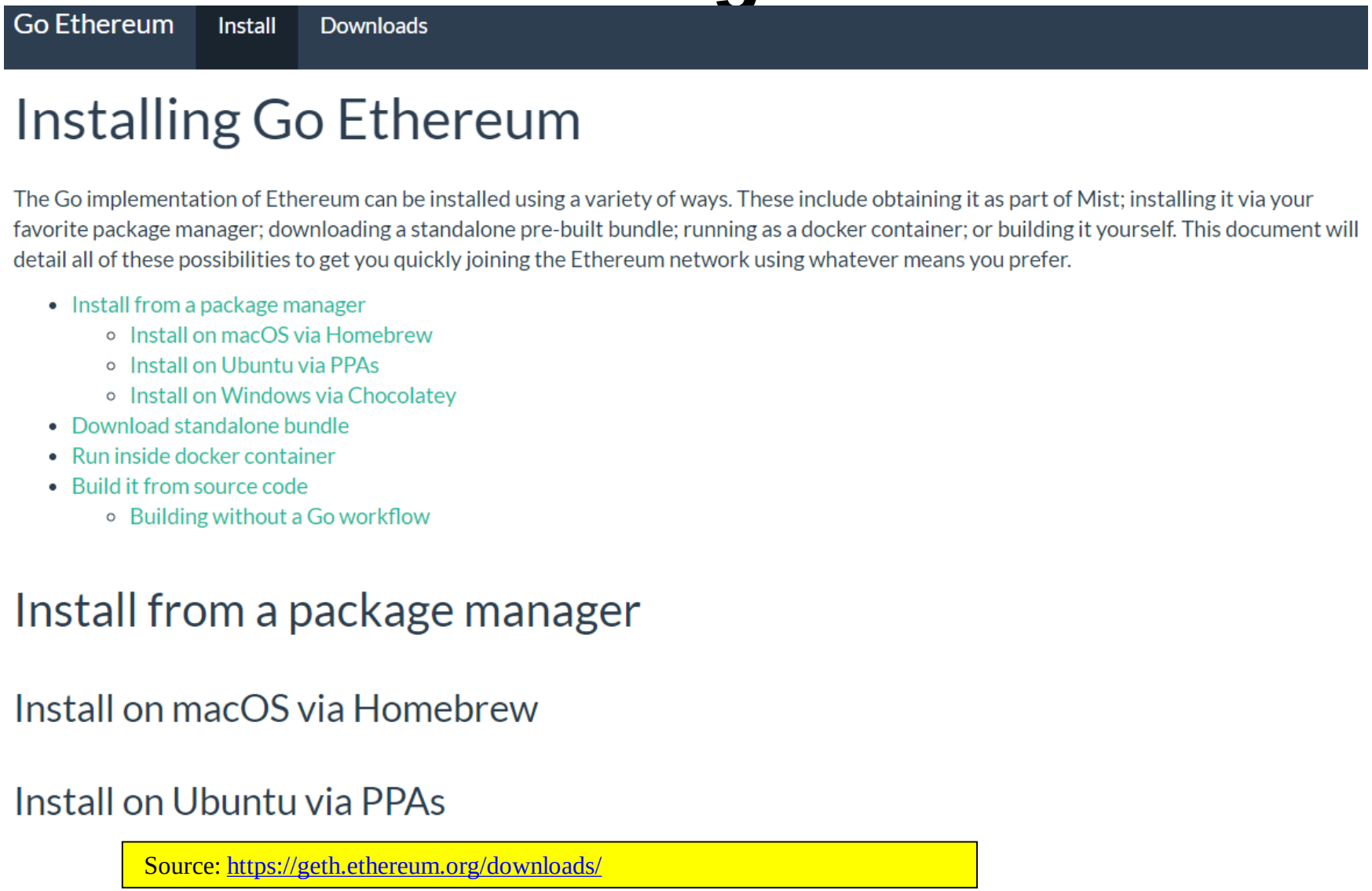
- All stable and develop builds of Geth and tools
- Archives for non-primary processor architectures
- Android library archives and iOS XCode frameworks

Please select your desired platform from the lists below and download your bundle of choice. Please be aware that the `MD5` checksums are provided by our binary hosting platform (Azure Blobstore) to help check for download errors. **For security guarantees please verify any downloads via the attached PGP signature files** (see OpenPGP Signatures for details).

Source: https://geth.ethereum.org/downloads/

Practical Exercise 04

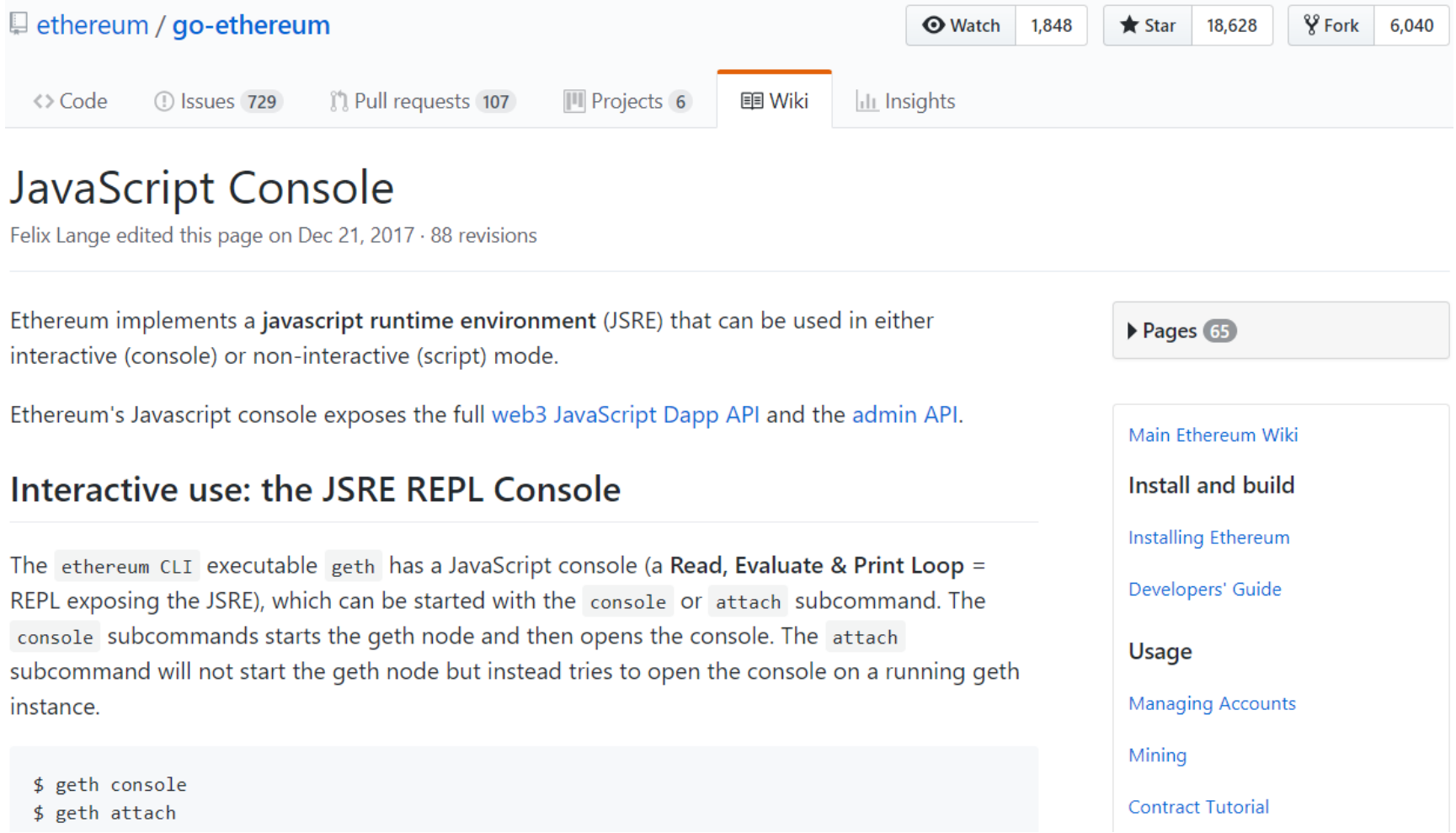
Installing Geth



Source: https://geth.ethereum.org/downloads/

Practical Exercise 04

Starting the Javascript Console



ethereum / go-ethereum

Watch 1,848 | Star 18,628 | Fork 6,040

Code | Issues 729 | Pull requests 107 | Projects 6 | Wiki | Insights

JavaScript Console

Felix Lange edited this page on Dec 21, 2017 · 88 revisions

Ethereum implements a **javascript runtime environment** (JSRE) that can be used in either interactive (console) or non-interactive (script) mode.

Ethereum's Javascript console exposes the full web3 JavaScript Dapp API and the admin API.

Interactive use: the JSRE REPL Console

The `ethereum CLI` executable `geth` has a JavaScript console (a **Read, Evaluate & Print Loop** = REPL exposing the JSRE), which can be started with the `console` or `attach` subcommand. The `console` subcommands starts the geth node and then opens the console. The `attach` subcommand will not start the geth node but instead tries to open the console on a running geth instance.

```
$ geth console
$ geth attach
```

Pages 65

Main Ethereum Wiki

Install and build

Installing Ethereum

Developers' Guide

Usage

Managing Accounts

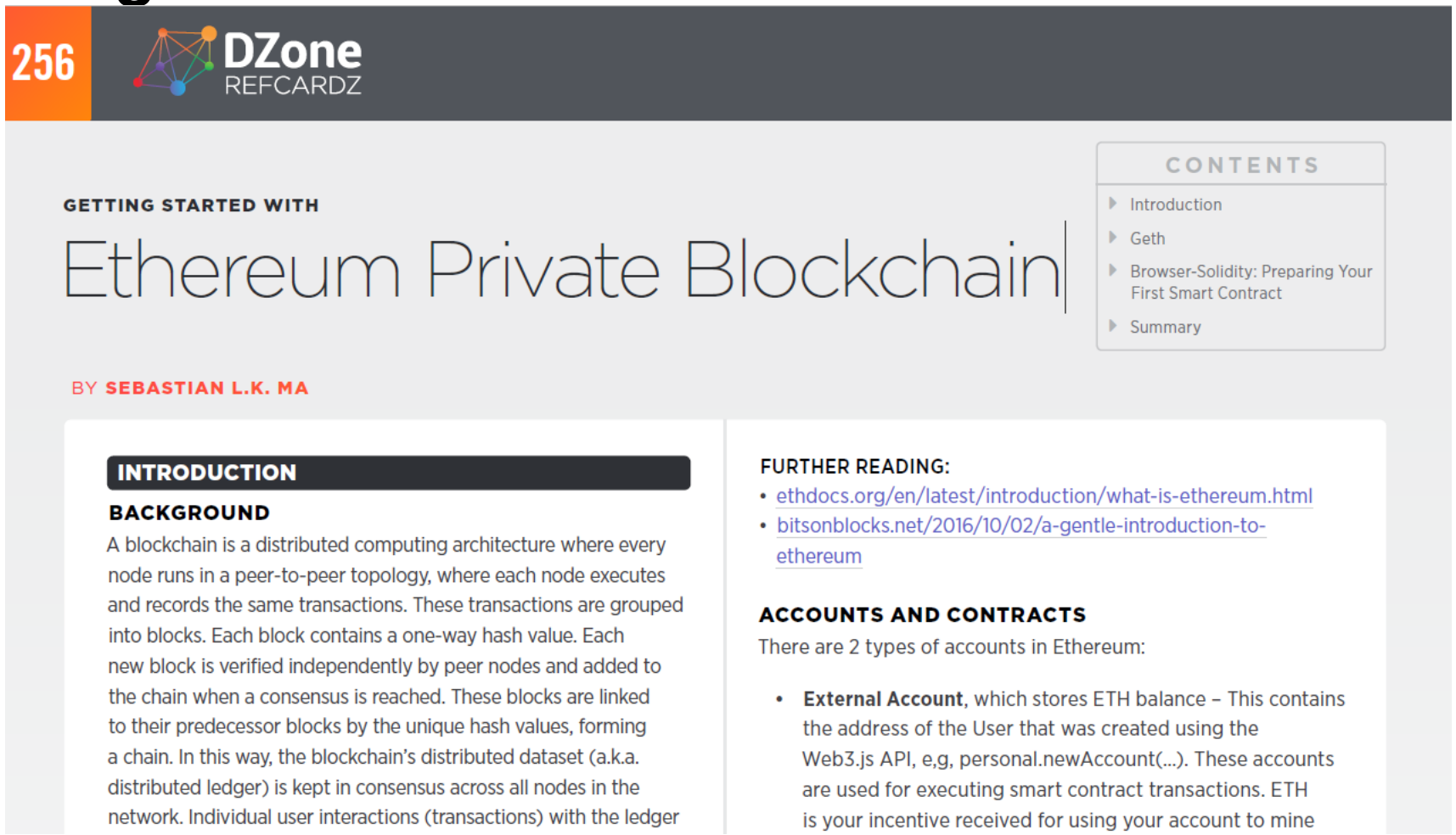
Mining

Contract Tutorial

Source: https://github.com/ethereum/go-ethereum/wiki/JavaScript-Console

Practical Exercise 04

Getting Started with Ethereum Private Blockchain



256 DZone REFCARDZ

GETTING STARTED WITH

Ethereum Private Blockchain

CONTENTS

BY SEBASTIAN L.K. MA

INTRODUCTION

BACKGROUND

A blockchain is a distributed computing architecture where every node runs in a peer-to-peer topology, where each node executes and records the same transactions. These transactions are grouped into blocks. Each block contains a one-way hash value. Each new block is verified independently by peer nodes and added to the chain when a consensus is reached. These blocks are linked to their predecessor blocks by the unique hash values, forming a chain. In this way, the blockchain's distributed dataset (a.k.a. distributed ledger) is kept in consensus across all nodes in the network. Individual user interactions (transactions) with the ledger

FURTHER READING:

- ethdocs.org/en/latest/introduction/what-is-ethereum.html
- bitsonblocks.net/2016/10/02/a-gentle-introduction-to-ethereum

ACCOUNTS AND CONTRACTS

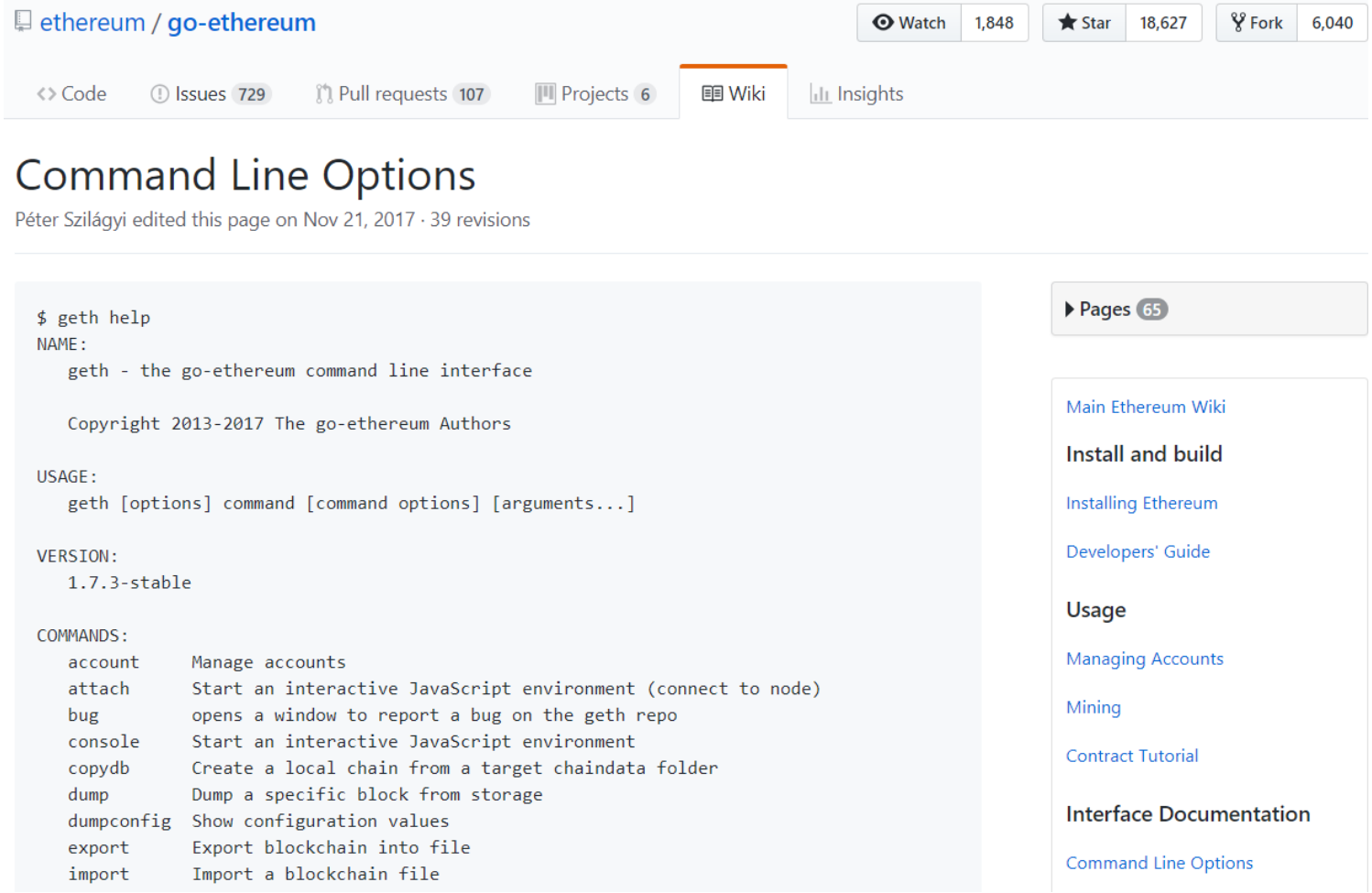
There are 2 types of accounts in Ethereum:

- **External Account**, which stores ETH balance – This contains the address of the User that was created using the Web3.js API, e,g, personal.newAccount(...). These accounts are used for executing smart contract transactions. ETH is your incentive received for using your account to mine

Source: https://dzone.com/refcardz/getting-started-with-etherium-private-blockchain?chapter=1/

Practical Exercise 04

Geth Command Line



ethereum / go-ethereum

Watch 1,848 | Star 18,627 | Fork 6,040

Code | Issues 729 | Pull requests 107 | Projects 6 | Wiki | Insights

Command Line Options

Péter Szilágyi edited this page on Nov 21, 2017 · 39 revisions

```
$ geth help
NAME:
   geth - the go-ethereum command line interface

   Copyright 2013-2017 The go-ethereum Authors

USAGE:
   geth [options] command [command options] [arguments...]

VERSION:
   1.7.3-stable

COMMANDS:
   account     Manage accounts
   attach      Start an interactive JavaScript environment (connect to node)
   bug         opens a window to report a bug on the geth repo
   console     Start an interactive JavaScript environment
   copydb      Create a local chain from a target chaindata folder
   dump        Dump a specific block from storage
   dumpconfig  Show configuration values
   export      Export blockchain into file
   import      Import a blockchain file
```

Pages 65

Main Ethereum Wiki

Install and build

Installing Ethereum

Developers' Guide

Usage

Managing Accounts

Mining

Contract Tutorial

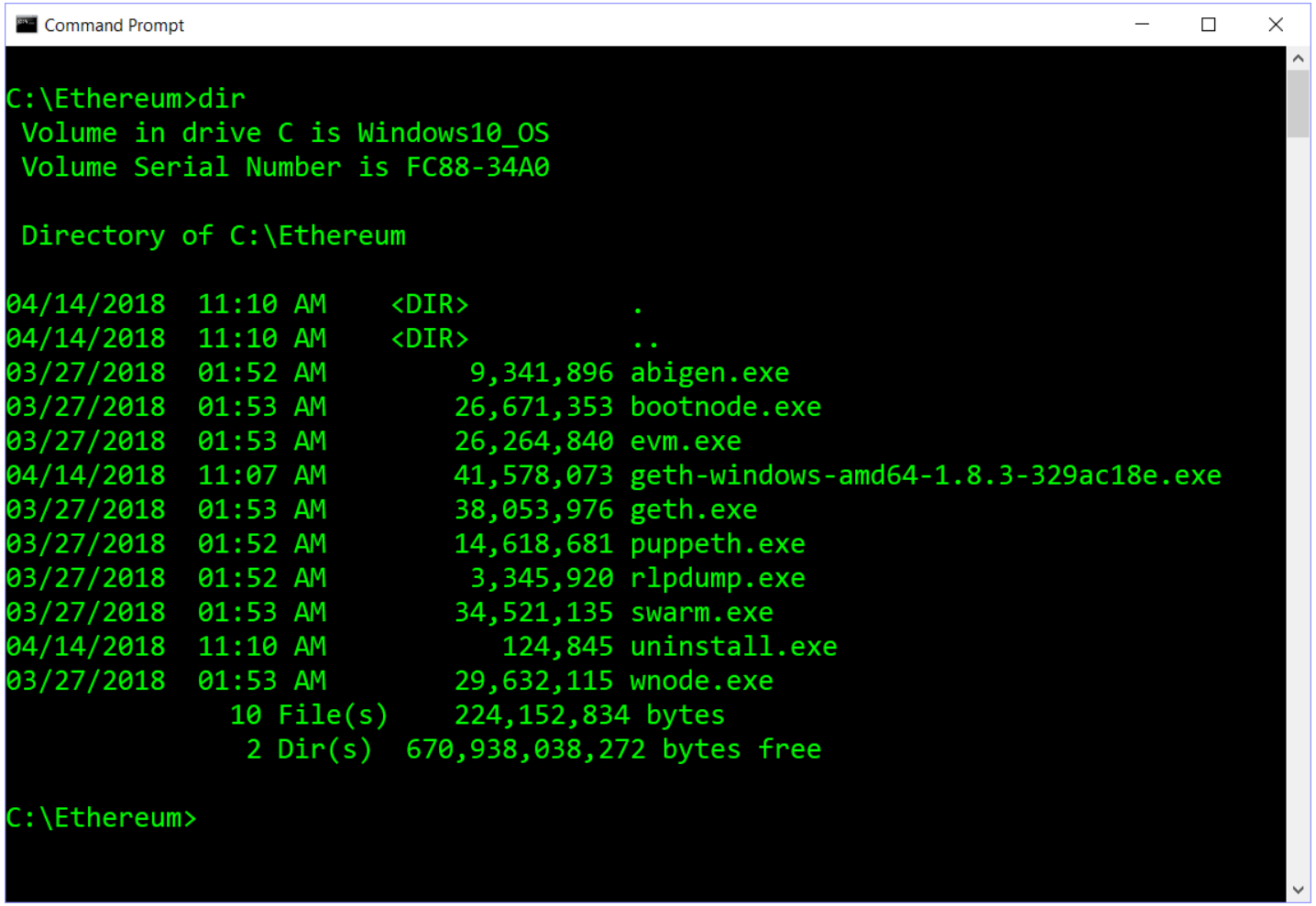
Interface Documentation

Command Line Options

Source: https://github.com/ethereum/go-ethereum/wiki/Command-Line-Options

Practical Exercise 04

In Windows, Geth at the Command



Practical Exercise 04

In Windows, Geth at the Command Line

To start Geth on the testnet , type this:

```
geth --testnet
```

You'll see text output similar to the screen in Figure 6-6, except that this mining is taking place on the testnet. Press Control+C to stop it.

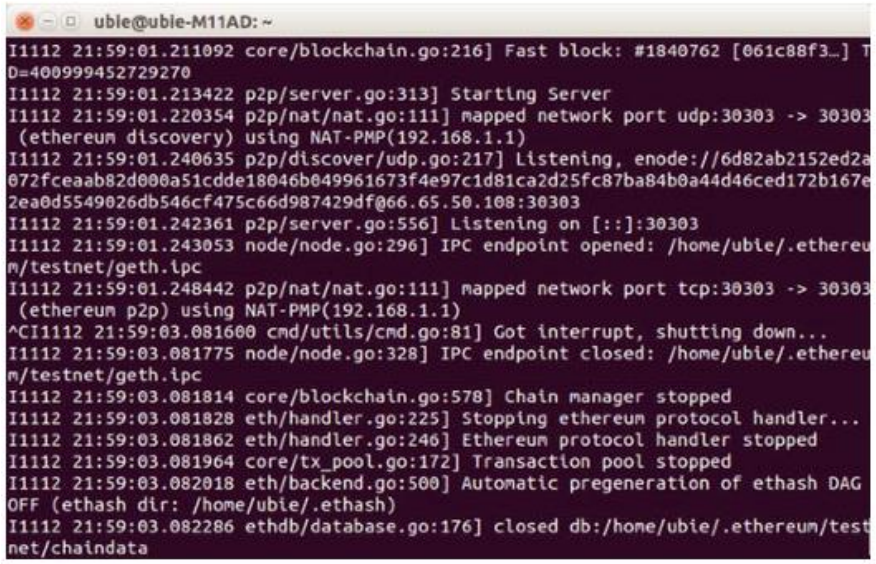


Figure 6-6. Output from testnet

Source: Introducing Ethereum and Solidity – by Chris Dannen (Published by Apress)

Practical Exercise 04

In Windows, Geth at the Command Line

For quick access to the CLI options, this short link is also available: http://cli.eth.guide .

As of this writing, network difficulty is fairly high, and solo miners might take a very long time to find a block. But in the next section, we'll start mining to our new wallet address anyway, to understand the experience of the miners who secure the network.

Fire Up Your Miner!

Geth does not begin mining automatically; you will give it the command to start or stop mining. In these examples, you will be mining with your machine's CPU. Mining with a GPU is more effective, but slightly more complicated, and is more suitable for specialized mining rigs anyway. We'll discuss these later in the chapter.

To begin mining on the main network, open a new Terminal window and enter the JavaScript console by typing the following:

Source: Introducing Ethereum and Solidity – by Chris Dannen (Published by Apress)

Practical Exercise 04

In Windows, Geth at the Command Line

```
geth console
```

You'll see the node begin to synchronize, but it will quickly return a command-line prompt where you can enter commands as Geth works in the background, so to speak.

> **Note**
> In the console, don't worry if the output text from mining or synchronization appears to overwrite your commands; it just appears that way. When you press Enter in the console, your command will be executed as normal, even if it seems to have broken onto several lines.

In order to get paid, you'll need to tell your node the Ethereum address for receiving your mining payments. Remember that because the EVM is a global virtual machine, it doesn't care whether the Ethereum address, or public key, you enter

Source: Introducing Ethereum and Solidity – by Chris Dannen (Published by Apress)

Practical Exercise 04

was created, or is currently associated with, your local computer. Everything is local to the EVM.

To set your etherbase as the recipient address for your payout, type this command in the console:

```
miner.setEtherbase(eth.accounts[your_address_here])
```

To finally begin mining, type this:

```
miner.start()
```

Boom! Your miner will begin. In the off-chance you find a block, your payment will be received at the address you set above, but don't be surprised if it takes days or even weeks. You'll see the node generating the DAG file and beginning the mining process , as shown in Figure 6-7. Why isn't ether mining an instant money-maker? That has a lot to do with your hardware, as you'll see below.

Source: https://github.com/ethereum/go-ethereum/wiki/JavaScript-Console

Practical Exercise 04

```
ubie@ubie-M11AD: ~
I1112 22:03:26.071880 eth/backend.go:454] Automatic pregeneration of ethash DAG
ON (ethash dir: /home/ubie/.ethash)
true
> I1112 22:03:26.072245 eth/backend.go:461] checking DAG (ethash dir: /home/ubie
/.ethash)
I1112 22:03:26.072435 miner/worker.go:539] commit new work on block 1748011 with
 0 txs & 0 uncles. Took 623.351µs
I1112 22:03:26.072570 ethash.go:259] Generating DAG for epoch 58 (size 156027865
0) (8f602dc7d86df0a7c8e7467ec0d211062ee85c5c14c6d2f6c025976cf550e8c5)
I1112 22:03:27.548451 ethash.go:291] Generating DAG: 0%
I1112 22:03:33.584568 ethash.go:291] Generating DAG: 1%
I1112 22:03:39.798725 ethash.go:291] Generating DAG: 2%
I1112 22:03:45.891413 ethash.go:291] Generating DAG: 3%
> I1112 22:03:51.758028 ethash.go:291] Generating DAG: 4%
> I1112 22:03:53.465117 eth/downloader/downloader.go:319] Block synchronisation
started
I1112 22:03:53.465561 miner/miner.go:75] Mining operation aborted due to sync op
eration
> I1112 22:03:57.340299 eth/downloader/downloader.go:298] Synchronisation failed
: receipt download canceled (requested)
```

Figure 6-7. The miner gets ready to mine

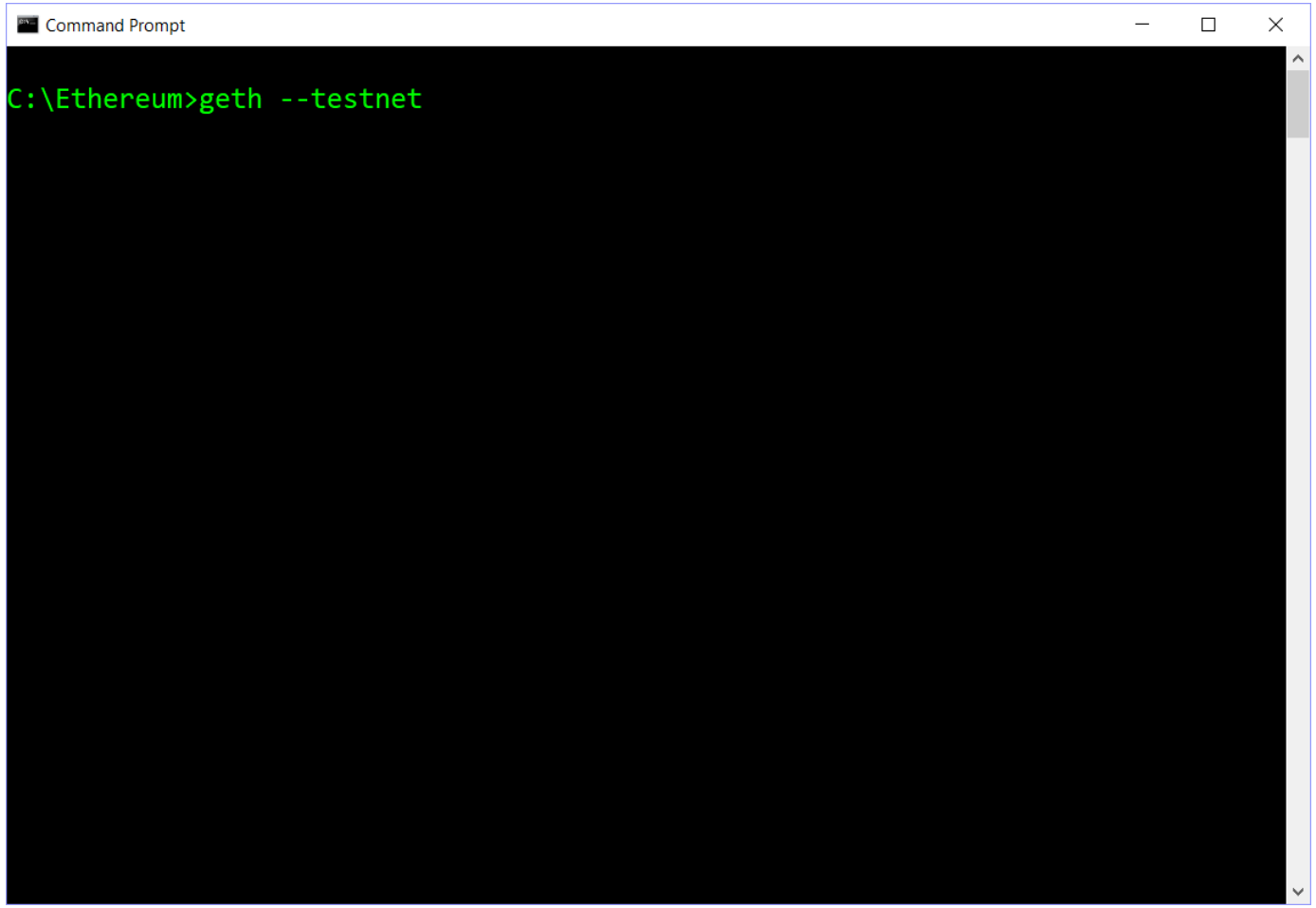
You can stop this process by typing the following:

```
miner.stop()
```

Next, you'll put a personal tag on the blocks you mine, just because.

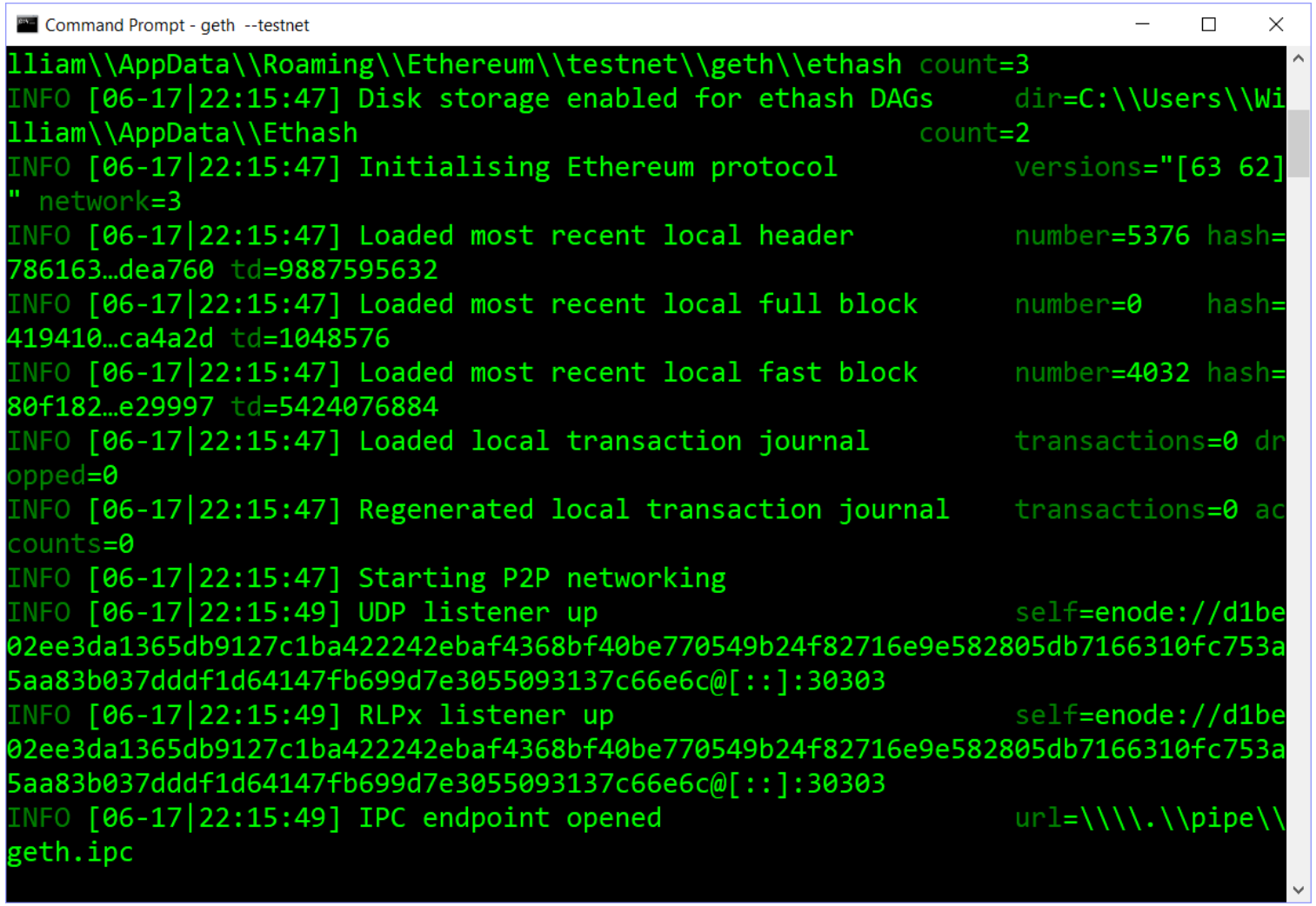
Source: https://github.com/ethereum/go-ethereum/wiki/JavaScript-Console

Practical Exercise 04



Source: https://github.com/ethereum/go-ethereum/wiki/JavaScript-Console

Practical Exercise 04



Source: https://github.com/ethereum/go-ethereum/wiki/JavaScript-Console

Practical Exercise 04

Exercise : Add Your Name to the Blockchain

Using the JavaScript console, you can add extra data—a grand total of 32 bytes, or enough to write some plain text or enter some ciphertext for someone else to read.

In the console, your miner should be stopped. Now type this JavaScript command with your name or a message between the quotes:

```
miner.setExtra("My_message_here")
```

Then type this:

```
miner.start()
```

The console will return true and begin mining. Should you find a block, it will be marked with your signature, which you can view on any blockchain explorer such as Etherchain (https://etherchain.org).

Source: https://github.com/ethereum/go-ethereum/wiki/JavaScript-Console

Practical Exercise 04

Exercise: Check Your Balance

Install the Web3.js library (https://github.com/ethereum/wiki/wiki/JavaScript-API#adding-web3) as described in the last section, to try out some of the Ethereum JavaScript API calls. These include checking a balance, sending a transaction, creating an account, and all sorts of other mathematical and blockchain-related functions. If your etherbase private key is held on your machine, for example, you can get the balance by typing in the console:

```
eth.getBalance(eth.coinbase).toNumber();
```

Hopefully by now, you have a working understanding of mining, and you've see it happen before your own eyes. In reality, the most effective way to see how mining moves state transition forward, executing contracts, is to work with the testnet.

Source: https://github.com/ethereum/go-ethereum/wiki/JavaScript-Console

Practical Exercise 04

Mining on the Testnet

One quick final note about mining. Recall in Chapter 5 that the Mist wallet can mine on the testnet, but not the main net. Why is this?

Actually, there is no need for Mist to mine on the main net and take up your computer's resources, because your contracts will execute without you mining. This is because there are currently thousands of nodes already mining on the public Ethereum chain, and being paid real ether to do so.

Note
If your contracts aren't executing on the testnet, don't go berserk! Turn your Mist or Geth testnet miner on, and your contracts will execute. This is a common mistake.

While there may coincidentally be others mining on the testnet while you are testing your contracts, there may also not be. Because there's no real financial incentive to leave a miner running on the testnet, you might find yourself in a lull, with nobody else on the testnet. This is why Mist allows testnet mining along with its GUI contract deployment interface.

Source: https://github.com/ethereum/go-ethereum/wiki/JavaScript-Console

Practical Exercise 05

Write a brief scenario that describes how Blockchain Technology would benefit your organization.

Practical Exercise 06

Using Analysis and Design Diagrams and Guidelines from the Lecture, write or describe in diagrams, a high-level scenario for a Blockchain application that could benefit your organization.

Practical Exercise 07

If you understand how Blockchain technologies could benefit your organization:

1. Write a brief plan how to deploy Blockchain Resources to make achieve your goals.

Or

2. Write a brief list of the things your organization needs to ramp up and get prepared to deploy Blockchain Technologies to help your organization achieve its Blockchain-related goals.

William Favre Slater, II

- 312-758-0307
- slater@billslater.com
- williamslater@gmail.com
- http://billslater.com/interview
- 1515 W. Haddon Ave., Unit 309
  Chicago, IL 60642
  United States of America

William Favre Slater, III