BlockchainWeekly

Powered by Shindig

What's Holding Blockchain Back from Mass Adoption? Quick Introduction to Blockchain Security & Blockchain Auditing

February 27, 2019

William Favre Slater, III

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

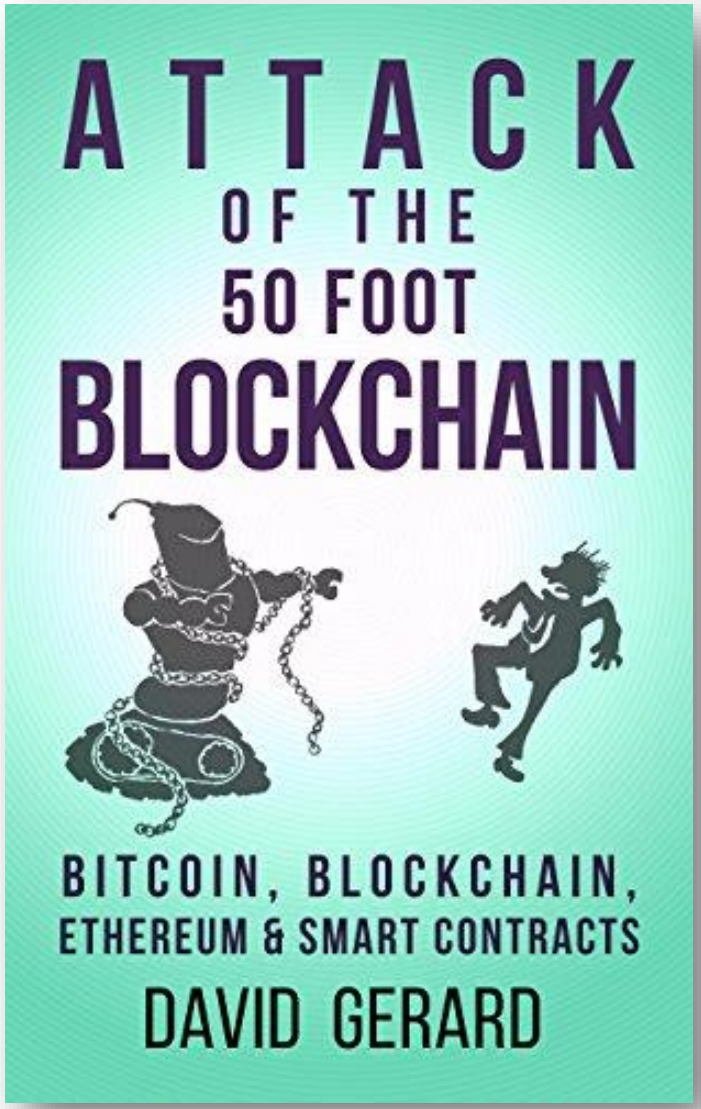
Presentation Location

http://billslater.com/writing

http://billslater.com

BlockchainWeekly

For a Cynical & Humorous View of Blockchain



BlockchainWeekly

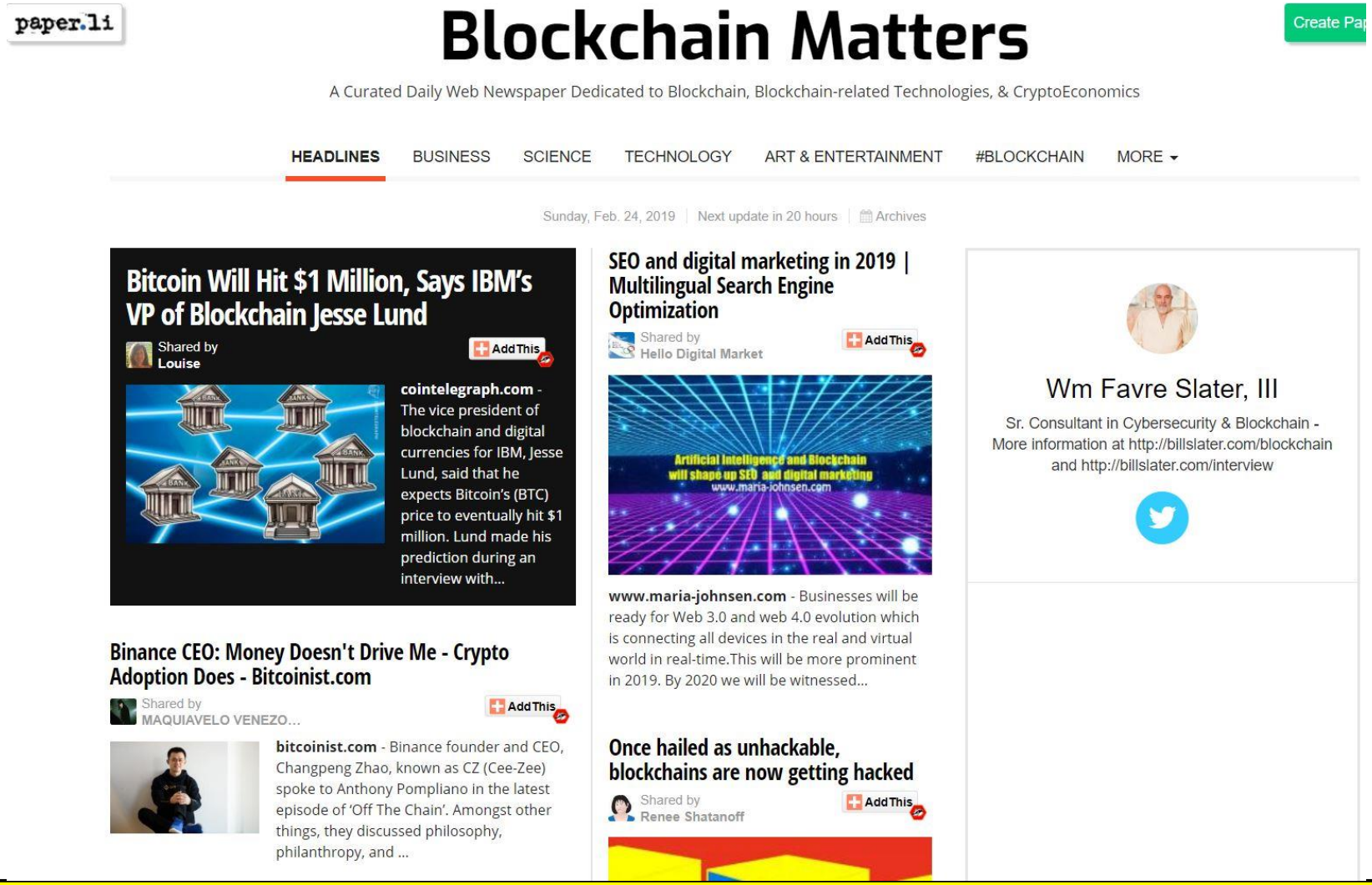
ISIS *Loves* Bitcoin (or They *Did* Love It)

Their Public Key

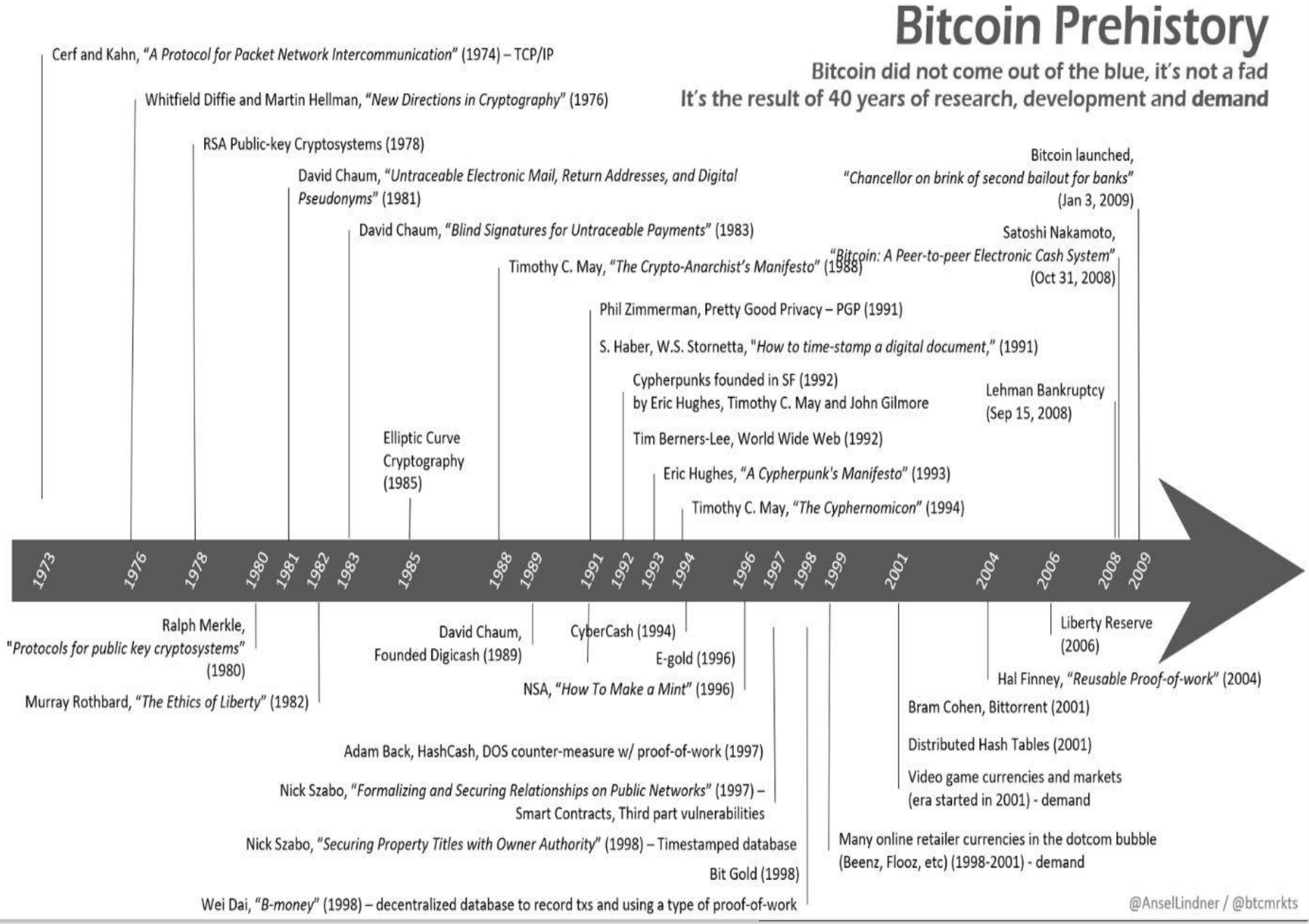
Comment:
This actually didn't end well for ISIS and their Donors.

Enough said.

New Free Blockchain Daily Newspaper:



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


Bitcoin Prehistory
Bitcoin did not come out of the blue, it's not a fad
It's the result of 40 years of research, development and demand
Cerf and Kahn, "A Protocol for Packet Network Intercommunication" (1974) – TCP/IP
Whitfield Diffie and Martin Hellman, "New Directions in Cryptography" (1976)
RSA Public-key Cryptosystems (1978)
David Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms" (1981)
David Chaum, "Blind Signatures for Untraceable Payments" (1983)
Timothy C. May, "The Crypto-Anarchist's Manifesto" (1988)
Phil Zimmerman, Pretty Good Privacy – PGP (1991)
S. Haber, W.S. Stornetta, "How to time-stamp a digital document," (1991)
Cypherpunks founded in SF (1992) by Eric Hughes, Timothy C. May and John Gilmore
Tim Berners-Lee, World Wide Web (1992)
Eric Hughes, "A Cypherpunk's Manifesto" (1993)
Timothy C. May, "The Cyphernomicon" (1994)
Elliptic Curve Cryptography (1985)
Bitcoin launched, "Chancellor on brink of second bailout for banks" (Jan 3, 2009)
Satoshi Nakamoto, "Bitcoin: A Peer-to-peer Electronic Cash System" (Oct 31, 2008)
Lehman Bankruptcy (Sep 15, 2008)
1973
1976
1978
1980
1981
1982
1983
1985
1988
1989
1991
1992
1993
1994
1996
1997
1998
1999
2001
2004
2006
2008
2009
Ralph Merkle, "Protocols for public key cryptosystems" (1980)
Murray Rothbard, "The Ethics of Liberty" (1982)
David Chaum, Founded Digicash (1989)
CyberCash (1994)
E-gold (1996)
NSA, "How To Make a Mint" (1996)
Liberty Reserve (2006)
Hal Finney, "Reusable Proof-of-work" (2004)
Bram Cohen, Bittorrent (2001)
Distributed Hash Tables (2001)
Video game currencies and markets (era started in 2001) - demand
Many online retailer currencies in the dotcom bubble (Beenz, Flooz, etc) (1998-2001) - demand
Adam Back, HashCash, DOS counter-measure w/ proof-of-work (1997)
Nick Szabo, "Formalizing and Securing Relationships on Public Networks" (1997) – Smart Contracts, Third part vulnerabilities
Nick Szabo, "Securing Property Titles with Owner Authority" (1998) – Timestamped database
Bit Gold (1998)
Wei Dai, "B-money" (1998) – decentralized database to record txs and using a type of proof-of-work
@AnselLindner / @btcmrkts

Abstract

Since Blockchain became well-known as the foundational set of technologies that enabled the creation and operation of Bitcoin, it has captured the attention and imagination of developers, industry leaders, and investors. This is because as a set of technologies that use consensus and peer-to-peer, decentralized systems, it creates immutable data records and enables trust and disintermediation at scale. So what is preventing Blockchain from changing the world?

This presentation will present some of the challenges that are preventing mass adoption of Blockchain, and some practical solutions to those challenges.

Agenda

What's Holding Back Blockchain from Mass Adoption?

Topic 1: Why Blockchain?
Topic 2: Blockchain Law
Topic 3: Distributed Systems and Blockchain Security Concepts
Topic 4: Blockchain Limits and Challenges
Topic 5: How to Secure Blockchain Infrastructure and Applications
Topic 6: How to perform Secure Software Development for Blockchain applications by design, coding practices, testing and verification
Topic 7: Blockchain and Auditing
Topic 8: How to Design and Implement a Blockchain Solution Project – an Organized High-Level Step-by-Step Approach
Topic 9: How to Help your Organization Rapidly Ramp Up Skills and Readiness for Blockchain Application Development
Conclusion
Special Thanks
References

TOPIC 1: WHY BLOCKCHAIN?

BlockchainWeekly

Powered by Shindig

Elements in favor of a blockchain approach

@rwang0 #Blockchain 10

Block chain use cases requires massive cloud resources

Establish trust

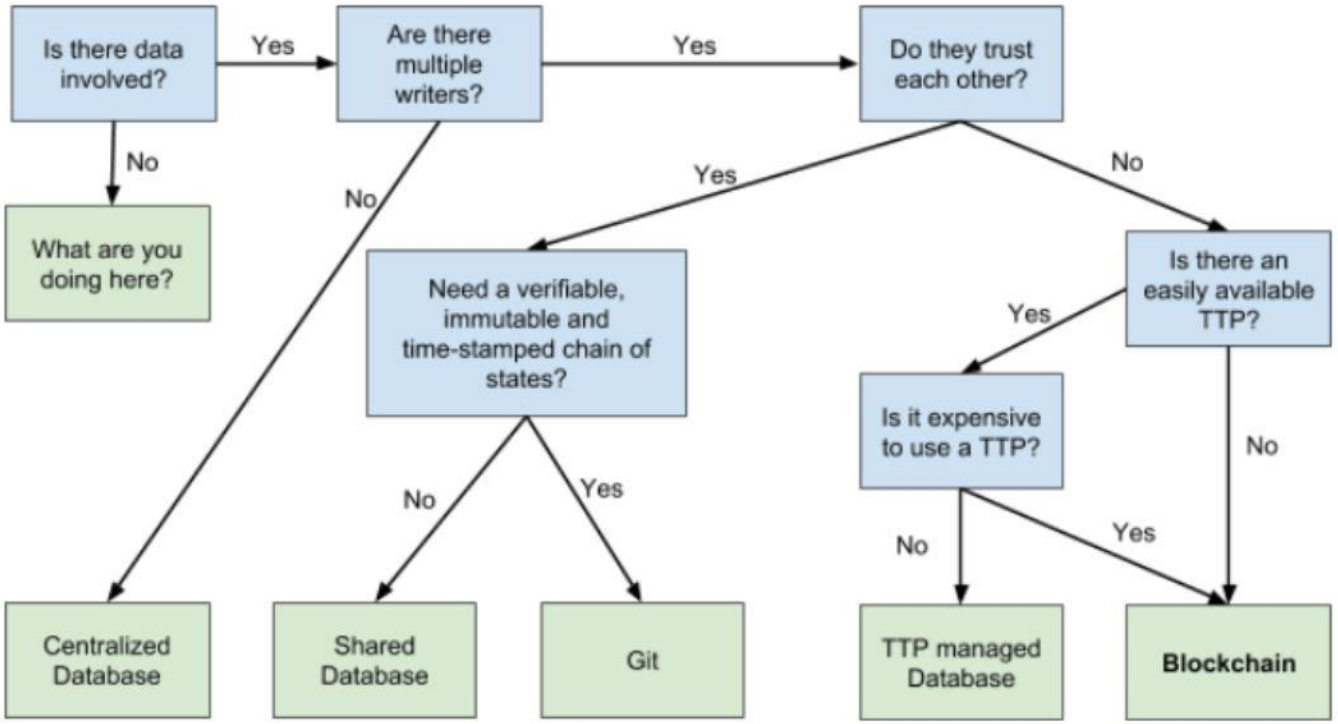
Transact on identity

Ensure provenance of data

Facilitate value exchange

Enable smart contracts

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:



Voila! You have now a framework to decide whether Blockchain technology is worth looking into. However, your journey doesn't end here. Once you figured out that a decentralized solution might be suited to your problem, there are kopp?

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

Blockchain Use Evolution

Defining Blockchain

A distributed ledger technology

Blockchain is a cryptographic, or encoded ledger – a database of transactions in the form of blocks arranged in a chain. These are validated by multiple users through consensus mechanisms (such as proof-of-work in Bitcoin mining) shared across a public or private network.

Blockchain technology could cut banks' infrastructure costs for cross-border payments, securities trading, and regulatory compliance

Potential benefits of Blockchain technology for the financial services industry

- Reduce costs of overall transactions and IT infrastructure
- Irrevocable and tamper-resistant transactions
- Reduction in systemic risks (eliminate credit and liquidity risks)
- Consensus in a variety of transactions
- Ability to store and define ownership of any tangible or intangible asset
- Increased accuracy of trade data and reduced settlement risk
- Near-instantaneous clearing and settlement
- Improved security and efficiency of transactions
- Enabling effective monitoring and auditing by participants, supervisors, and regulators

2009-2012 Foundation days

- Emergence of Bitcoin based on a paper by Satoshi Nakamoto
- On January 3, 2009, the Genesis block was mined
- Experimental and limited to cryptographic community
- Blockchain as the backbone of Bitcoin

2012-2014 Moving beyond the cryptographers

- Rise of Bitcoin exchanges
- Mixed response to Bitcoin as it struggles with money laundering and criminal activity, but also gains acceptance across some online retail stores among others
- Rise of Bitcoin- based startups
- Bitcoin price surged to US$1,000
- Blockchain gains attention of financial services firms (begins internal trials)

2014-2015 Blockchain buzz years

- Blockchain, the underlying technology behind Bitcoin, gets serious attention and investment from financial services firms, regulators, and VCs
- Explosion of use cases within BFSI
- Announcement of consortiums to accelerate adoption, innovation, and common standards
- Banks experiment with their versions of cryptocurrencies
- Global service providers and technology companies put their weight behind Blockchain

2016-2017 Crossing the chasm

- The next two years are critical for Blockchain technology to demonstrate sustainable value and show adoption beyond proofs of concept by FS firms
- Startups backed by VC funding and consortiums need to show results to justify the large sums of funding and/or investment of time and resources
- Scalability and throughput issues need to be solved for the Blockchain technology to cross the chasm to mainstream adoption

2018-2020 Adoption movement

- Consortiums will be instrumental in defining protocols and common standards to facilitate widespread adoption
- Regulatory bodies likely to play a key role in facilitating adoption while ensuring compliance
- Explosion of use cases beyond BFSI
- IT service providers likely to accelerate investments to build capabilities around Blockchain technology implementation
- Rise of IPOs and Unicorns in the Blockchain startup ecosystem

2020 & beyond Accelerated adoption

- Blockchain will gain adoption within and beyond BFSI, leading to new business models at the intersection of advanced analytics, IoT, and Blockchain based smart contracts
- Blockchain is referenced in two major shifts expected to occur in the nearest future, according to a report by World Economic Forum: The first tax collected by government using the Blockchain technology by 2023. The second one is storing more than 10% of global gross domestic product in Blockchains by 2027
- Banks' infrastructure costs for cross-border payments, securities trading, and regulatory compliance reduced by US$15-20 billion a year from 2022, according to a recent report by Spanish bank Santander

Why Is Blockchain Important?

- Accessible
- Open source
- Easily provides three challenging elements of the **Parkerian Hexad** model for security:
  - **Authenticity**
  - **Control**
  - **Utility**
- 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

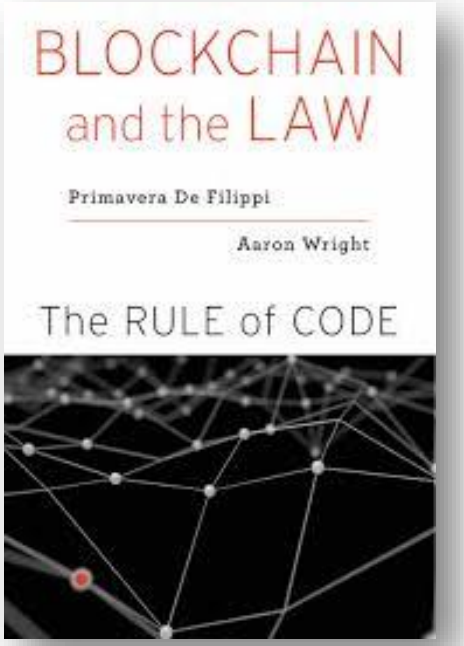
TOPIC 2: BLOCKCHAIN LAW

BlockchainWeekly

Powered by Shindig

Blockchain & 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.



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

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

When Blockchains Crash, Who Can We Sue?

757 views | Feb 7, 2019, 02:32pm

When Blockchains Crash, Who Can You Sue?

Andrea Tinianow Contributor ⓘ
Crypto & Blockchain
I am the Blockchain Czarina. I bring you the world of blockchain.

GETTY

Delaware corporate law is rich in rules arising from issues of trust and the application of fiduciary duties. Usually the rules relate to whether the directors of a corporate board have breached their fiduciary duty of care or loyalty to the company or shareholders. While this framework affords directors considerable leeway to manage the affairs of the company through a bedrock principle of Delaware law called the business judgment rule, it also serves to deter directors from engaging in problematic behavior and to hold directors responsible when they act carelessly or put their own interests above those of the shareholders.

Published February 7, 2019 at Forbes.com

Source: https://www.forbes.com/sites/andreatinianow/2019/02/07/when-blockchains-crash-whom-can-you-sue/#760e20707775

Blockchain & The Law

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

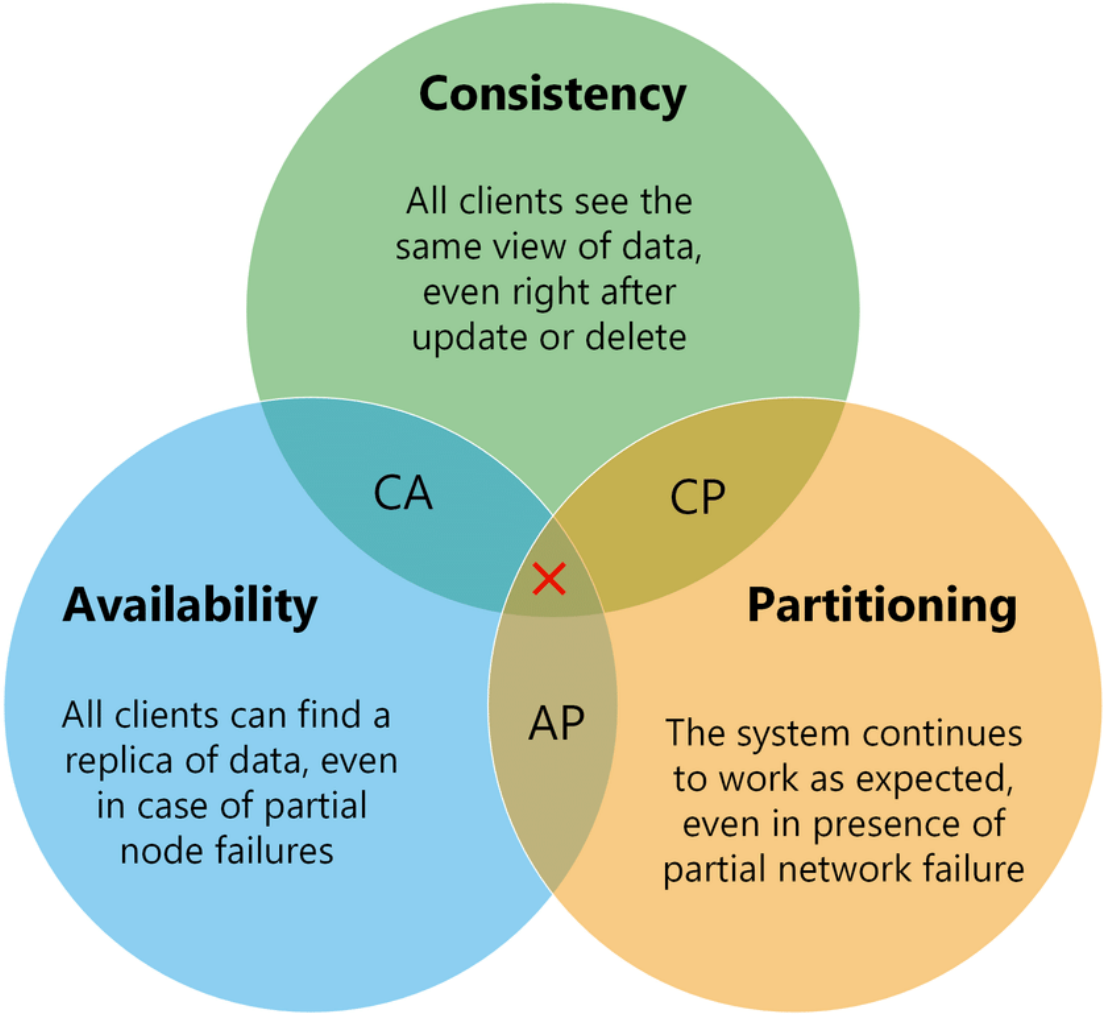
Ms. Puneet Bhasin
Mumbai, India
https://www.linkedin.com/in/advpuneetbhasincyberlawyer/

BlockchainWeekly

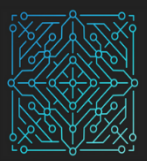
TOPIC 3: DISTRIBUTED SYSTEMS AND BLOCKCHAIN SECURITY CONCEPTS

BlockchainWeekly

CAP Theorem

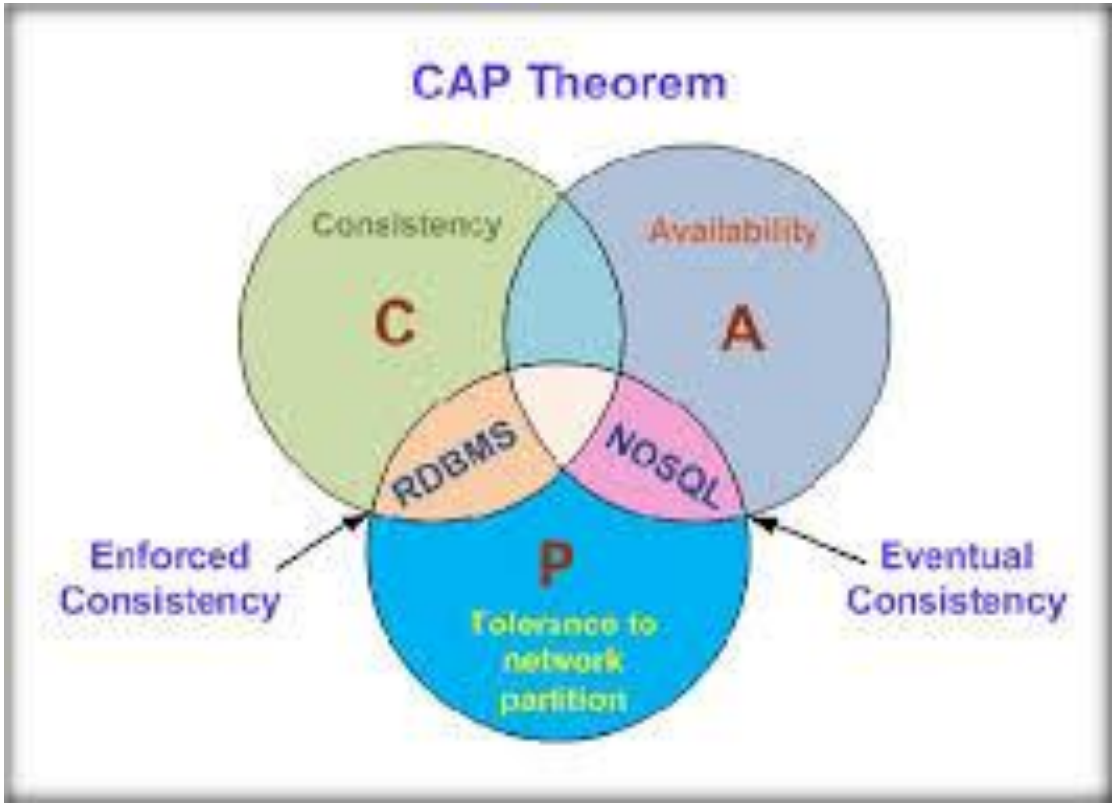


Source: https://en.wikipedia.org/wiki/CAP_theorem



BlockchainWeekly

CAP Theorem



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

CAP Theorem

CAP Theorem

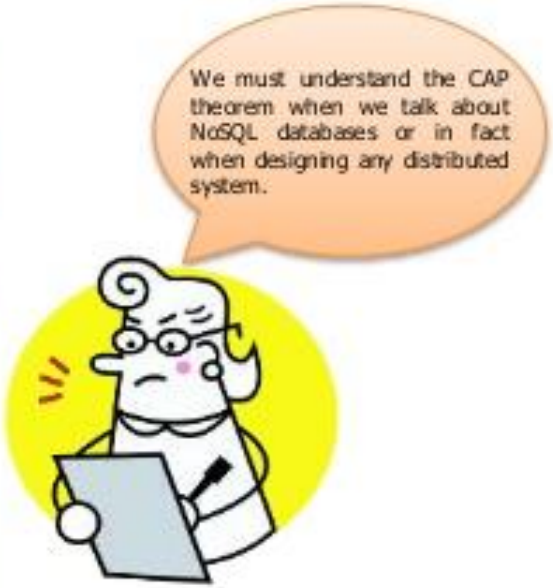
edureka!

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

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

Blockchain Security Threats and Vulnerabilities & Remediations (A Short List - 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 & Remediations (A Short List - 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 & Remediations (A Short List - 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**

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

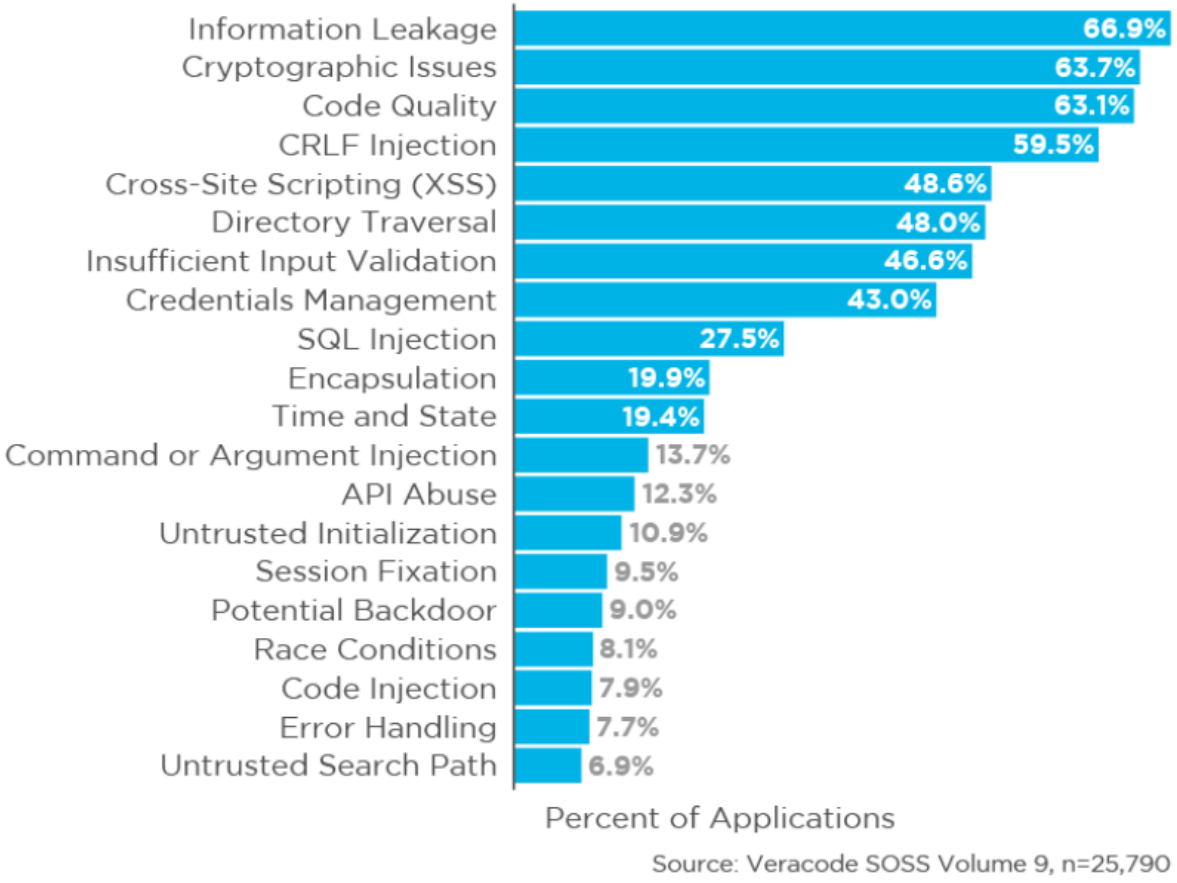
51% Attack on Ethereum Classic - January 2019

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

BlockchainWeekly

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 4: BLOCKCHAIN LIMITS AND CHALLENGES

BlockchainWeekly

Powered by Shindig

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.

BlockchainWeekly

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.

BlockchainWeekly

Limits and 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?

TOPIC 5: HOW TO SECURE BLOCKCHAIN INFRASTRUCTURE AND APPLICATIONS

BlockchainWeekly

Powered by Shindig

Topic 5: How to Secure Blockchain Infrastructure and Applications

- Build and lead Teams of experienced, dedicated workers
- Design securely
- 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

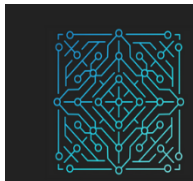
TOPIC 6: HOW TO PERFORM SECURE SOFTWARE DEVELOPMENT FOR BLOCKCHAIN APPLICATIONS BY DESIGN, CODING PRACTICES, TESTING AND VERIFICATION

BlockchainWeekly

Powered by Shindig

Topic 6: How to Perform Secure Software Development for Blockchain applications by Design, Coding practices, Testing and Verification

- Experienced DApp developers
- Test-driven Development
- 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



CASE STUDIES

BlockchainWeekly

Powered by Shindig

Case Study 1

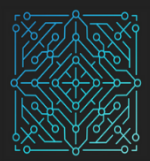
- Timeframe: November 2017
- Location: User ***devops199*** somewhere on the Ethereum Blockchain
- Topic: Placement in Production of flawed Smart Contract
- Results: Loss of over $150 million

$150,000,000 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
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    }
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      m_spentToday = 0;
212    }
213
214 +  // throw unless the contract is not yet initialized.
215 +  modifier only_uninitialized { if (m_numOwners > 0) throw; _; }
```



BlockchainWeekly

Shindig

TOPIC 7: BLOCKCHAIN AND AUDITING

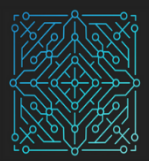
BlockchainWeekly

Topic 7: Blockchain and Auditing

- Blockchain Integrity and Security
- DApps
- Infrastructure
- Physical Security

Concepts of Auditing the Data and Transactions in Blockchain Data Structures

- Data should be validated and verified prior to committing as a Blockchain transaction because once written to the Blockchain it is immutable.
- Sample transactions should be verified from the DApp, to as successfully written to the Blockchain.

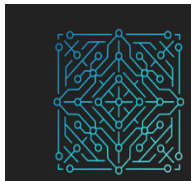


BlockchainWeekly

Automating the Auditing of Blockchains and Blockchain Applications

- In February 2018, ***Maian***, an open source tool to monitor Smart Contracts for being Greedy, Prodigal, or Suicidal was announced.

- As of April 2018, EY has Blockchain Auditing tools and technology.
  - https://www.ey.com/en_gl/news/2018/04/ey-announces-blockchain-audit-technology
- As of October 2018, How Big Four Auditors Delve Into Blockchain: PwC, Deloitte, EY and KPMG Approaches Compared
  - https://cointelegraph.com/news/how-big-four-auditors-delve-into-blockchain-pwc-deloitte-ey-and-kpmg-approaches-compared



BlockchainWeekly

Auditing Smart Contracts at Scale

Finding The Greedy, Prodigal, and Suicidal Contracts at Scale

Ivica Nikolić
School of Computing, NUS
Singapore

Aashish Kolluri
School of Computing, NUS
Singapore

Ilya Sergey
University College London
United Kingdom

Prateek Saxena
School of Computing, NUS
Singapore

Aquinas Hobor
Yale-NUS College and School of Computing, NUS
Singapore

Abstract

Smart contracts—stateful executable objects hosted on blockchains like Ethereum—carry billions of dollars worth of coins and cannot be updated once deployed. We present a new systematic characterization of a class of *trace vulnerabilities*, which result from analyzing multiple invocations of a contract over its lifetime. We focus attention on three example properties of such trace vulnerabilities: finding contracts that either lock funds indefinitely, leak them carelessly to arbitrary users, or can be killed by anyone. We implemented MAIAN, the first tool for precisely specifying and reasoning about trace properties, which employs inter-procedural symbolic analysis and concrete validator for exhibiting real exploits. Our analysis of nearly one million contracts flags 34,200 (2,365 distinct) contracts vulnerable, in 10 seconds per contract. On a subset of 3,759 contracts which we sampled for concrete validation and manual analysis, we reproduce real exploits at a true positive rate of 89%, yielding exploits for 3,686 contracts. Our tool finds exploits for the infamous Parity bug that indirectly locked 200 million dollars worth in Ether, which previous analyses failed to capture.

1 Introduction

Cryptocurrencies feature a distributed protocol for a set of computers to agree on the state of a public ledger purpose applications. Contracts are programs that run on blockchains: their code and state is stored on the ledger, and they can send and receive coins. Smart contracts have been popularized by the Ethereum blockchain. Recently, sophisticated applications of smart contracts have arisen, especially in the area of token management due to the development of the ERC20 token standard. This standard allows the uniform management of custom tokens, enabling, *e.g.*, decentralized exchanges and complex wallets. Today, over a million smart contracts operate on the Ethereum network, and this count is growing.

Smart contracts offer a particularly unique combination of security challenges. Once deployed they cannot be upgraded or patched,[1] unlike traditional consumer device software. Secondly, they are written in a new ecosystem of languages and runtime environments, the de facto standard for which is the Ethereum Virtual Machine and its programming language called Solidity. Contracts are relatively difficult to test, especially since their runtimes allow them to interact with other smart contracts and external off-chain services; they can be invoked repeatedly by transactions from a large number of users. Third, since coins on a blockchain often have significant value, attackers are highly incentivized to find and exploit bugs in contracts that process or hold them directly for profit. The attack on the DAO contract cost the Ethereum community $60 million US; and several more recent ones have had impact of a similar scale [1].

In this work, we present a systematic characterization

February 2018 Technical paper about flaws in
How Ethereum and EVM handle Smart Contracts. Worth your time.

Prodigal - Leak them carelessly to arbitrary users

Suicidal - Can be killed by anyone

Greedy - Lock funds Indefinitely

Source: https://www.reddit.com/r/Bitcoin/comments/7ys5nq/pdf_finding_the_greedy_prodigal_and_suicidal/

Auditing Smart Contracts at Scale

Finding The Greedy, Prodigal, and Suicidal Contracts at Scale

5.4 Summary and Observations

The symbolic execution engine of MAIAN flags 34,200 contracts. With concrete validation engine or manual inspection, we have confirmed that around 97% of prodigal, 97% of suicidal and 69% of greedy contracts are true positive. The importance of analyzing the bytecode of the contracts, rather than Solidity source code, is demonstrated by the fact that only 1% of all contracts have source code. Further, among all flagged contracts, only 181 have verified source codes according to the widely used platform Etherscan, or in percentages only 1.06%, 0.47% and 0.49%, in the three categories of prodigal, suicidal, and greedy, respectively. We refer the reader to Table 1 for the exact summary of these results.

Furthermore, the maximal amount of Ether that could have been withdrawn from prodigal and suicidal contracts, before the block height BH, is nearly 4,905 Ether, or 5.9 million US dollars[10] according to the exchange rate at the time of this writing. In addition, 6,239 Ether (7.5 million US dollars) is locked inside posthumous contracts currently on the blockchain, of which 313 Ether (379,940 US dollars) have been sent to dead contracts after they have been killed.

Finally, the analysis given in Table 2 shows the number of flagged contracts for different invocation depths from 1 to 4. We tested 25,000 contracts being for greedy, and 100,000 for remaining categories, inferring that increasing depth improves results marginally, and an invocation depth of 3 is an optimal tradeoff point.

Inv. depth	Prodigal	Suicidal	Greedy
1	131	127	682
2	156	141	682
3	157	141	682
4	157	141	682

Table 2: The table shows number of contracts flagged for various invocation depths. This analysis is done on a random subset of 25,000–100,000 contracts.

7 Conclusion

We characterize vulnerabilities in smart contracts that are checkable as properties of an entire execution trace (possibly infinite sequence of their invocations). We show three examples of such trace vulnerabilities, leading to greedy, prodigal and suicidal contracts. Analyzing 970,898 contracts, our new tool MAIAN flags thousands of contracts vulnerable at a high true positive rate.

Prodigal - Leak them carelessly to arbitrary users

Suicidal - Can be killed by anyone

Greedy - Lock funds Indefinitely

Bottom Line: three to four percent of the smart contracts on Ethereum's blockchain still contain trace vulnerabilities, according to the researchers' new analysis methodology.

Sources: https://www.reddit.com/r/Bitcoin/comments/7ys5nq/pdf_finding_the_greedy_prodigal_and_suicidal/ and https://bitsonline.com/singapore-research-ethereum/

Auditing Smart Contracts at Scale

Finding The Greedy, Prodigal, and Suicidal Contracts at Scale

Opacity Is Hampering Ethereum Security

Another interesting point raised in the paper is the unavailability of smart contract source code for Ethereum smart contracts, estimating the number at only one percent of the 970 thousand contracts they analyzed.

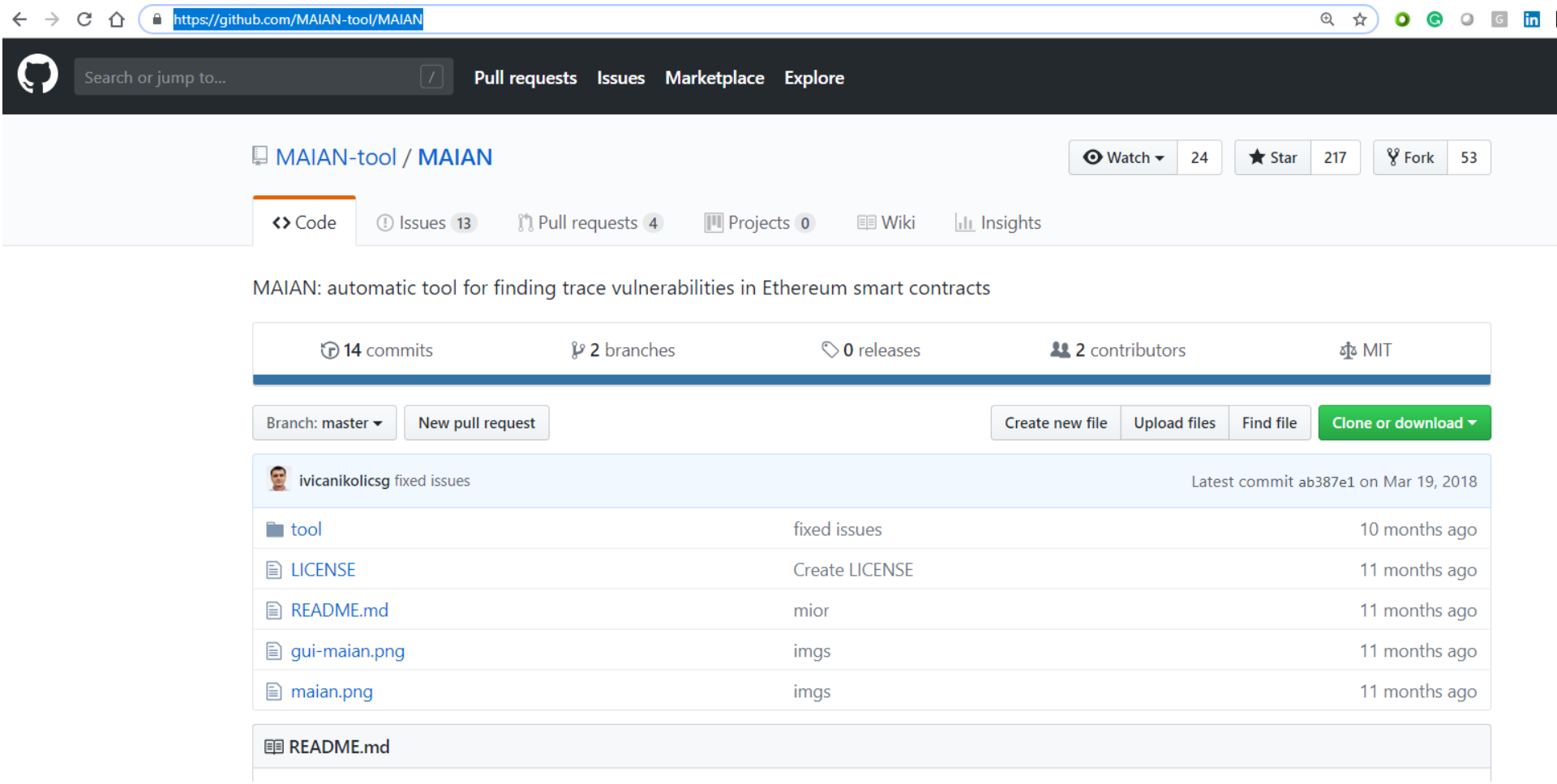
Fixing serious security vulnerabilities at scale requires **peer review**, and the **culture of propriety on the Ethereum network** forced the research team to directly analyze EVM bytecode instead of the sources to complete their research. Were the source code for these contracts more available and reviewed, Trace Vulnerabilities on Ethereum may not have proliferated in the first place.

-- 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.

Bottom Line: three to four percent of the smart contracts on Ethereum's blockchain still contain trace vulnerabilities, according to the researchers' new analysis methodology.

Sources: https://www.reddit.com/r/Bitcoin/comments/7ys5nq/pdf_finding_the_greedy_prodigal_and_suicidal/ and https://bitsonline.com/singapore-research-ethereum/

MAIAN



Source https://github.com/MAIAN-tool/MAIAN

MAIAN

Maian

The repository contains Python implementation of Maian -- a tool for automatic detection of buggy Ethereum smart contracts of three different types: prodigal, suicidal and greedy. Maian processes contract's bytecode and tries to build a trace of transactions to find and confirm bugs. The technical aspects of the approach are described in our paper.

Evaluating Contracts

Maian analyzes smart contracts defined in a file `<contract file>` with:

1. Solidity source code, use `-s <contract file> <main contract name>`
2. Bytecode source, use `-bs <contract file>`
3. Bytecode compiled (i.e. the code sitting on the blockchain), use `-b <contract file>`

Maian checks for three types of buggy contracts:

1. Suicidal contracts (can be killed by anyone, like the Parity Wallet Library contract), use `-c 0`
2. Prodigal contracts (can send Ether to anyone), use `-c 1`
3. Greedy contracts (nobody can get out Ether), use `-c 2`

For instance, to check if the contract `ParityWalletLibrary.sol` given in Solidity source code with `WalletLibrary` as main contract is suicidal use

```
$ python maian.py -s ParityWalletLibrary.sol WalletLibrary -c 0
```

Source https://github.com/MAIAN-tool/MAIAN

MAIAN

The output should look like this:

```
================================================================================================
[ ] Compiling Solidity contract from the file example_contracts/ParityWalletLibrary.sol ...  Done
[ ] Connecting to PRIVATE blockchain emptychain  ... ESTABLISHED
[ ] Deploying contract .... confirmed at address: 0x9E536236ABF2288a7864C6A1AfaA4Cb98D464306
[ ] Contract code length on the blockchain :    16530  : 0x6060604052600436106101d5760...
[ ] Contract address saved in file: ./out/WalletLibrary.address
[ ] Check if contract is SUICIDAL

[ ] Contract address    : 0x9E536236ABF2288a7864C6A1AfaA4Cb98D464306
[ ] Contract bytecode   : 6060604052600436106101d576000357c0100000000000000...
[ ] Bytecode length     : 16528
[ ] Blockchain contract: True
[ ] Debug               : False

[ ] Search with call depth: 1   : 11111111111111111111
[ ] Search with call depth: 2   : 11122222222222222222222212222222222222222222212222

[-] Suicidal vulnerability found!

    The following 2 transaction(s) will trigger the contract to be killed:
    -Tx[1] :e46dcfeb 0000000000000000000000000000000000000000000000000000000000000040 0000000000000000
0000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000000000000000
0000000000000
    -Tx[2] :cbf0b0c0

    The transactions correspond to the functions:
    -initWallet(address[],uint256,uint256)
    -kill(address)

[ ] Confirming suicide vulnerability on private chain ... ..... tx[0] mined ........ tx[1] mined
    Confirmed ! The contract is suicidal !
```

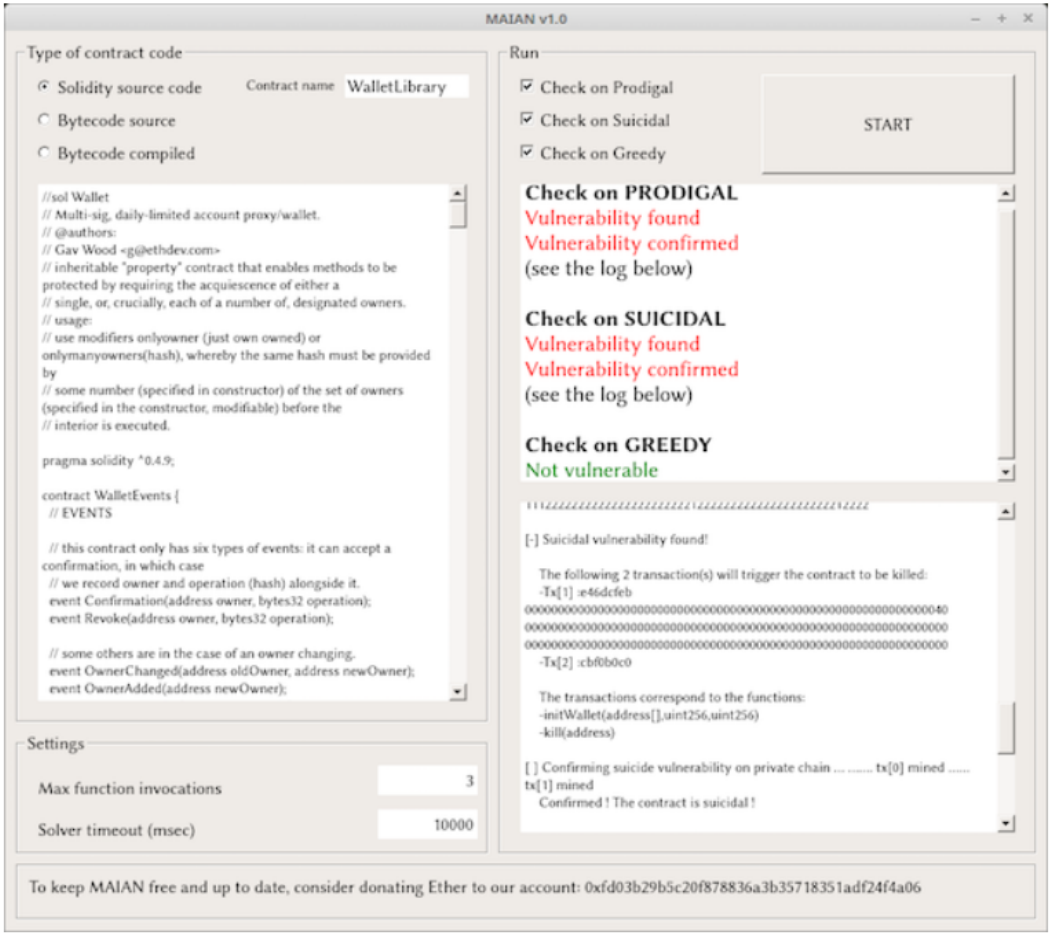
To get the full list of options use `python maian.py -h`

Source https://github.com/MAIAN-tool/MAIAN

MAIAN

GUI

For GUI inclined audience, we provide a simple GUI-based Maian. Use `python gui-maian.py` to start it. A snapshot of one run is given below



Source https://github.com/MAIAN-tool/MAIAN

MAIAN

Supported Operating Systems and Dependencies

Maian should run smoothly on Linux (we've checked on Ubuntu/Mint) and MacOS. Our attempts to run it on Windows have failed. The list of dependencies is as follows:

1. Go Ethereum, check https://ethereum.github.io/go-ethereum/install/
2. Solidity compiler, check http://solidity.readthedocs.io/en/develop/installing-solidity.html
3. Z3 Theorem prover, check https://github.com/Z3Prover/z3
4. web3, try `pip install web3`
5. PyQt5 (only for GUI Maian), try `sudo apt install python-pyqt5`

Important

To reduce the number of false positives, Maian deploys the analyzed contracts (given either as Solidity or bytecode source) on a private blockchain, and confirms the found bugs by sending appropriate transactions to the contracts. Therefore, during the execution of the tool, a private Ethereum blockchain is running in the background (blocks are mined on it in the same way as on the Mainnet). Our code stops the private blockchain once Maian finishes the search, however, in some extreme cases, the blockchain keeps running. Please make sure that after the execution of the program, the private blockchain is off (i.e. `top` does not have `geth` task that corresponds to the private blockchain).

License

Maian is released under the MIT License, i.e. free for private and commercial use.

Source https://github.com/MAIAN-tool/MAIAN

TOPIC 8: HOW TO DESIGN AND IMPLEMENT A BLOCKCHAIN SOLUTION PROJECT – AN ORGANIZED HIGH-LEVEL STEP-BY-STEP APPROACH

BlockchainWeekly

Powered by Shindig

Overview of Ethereum

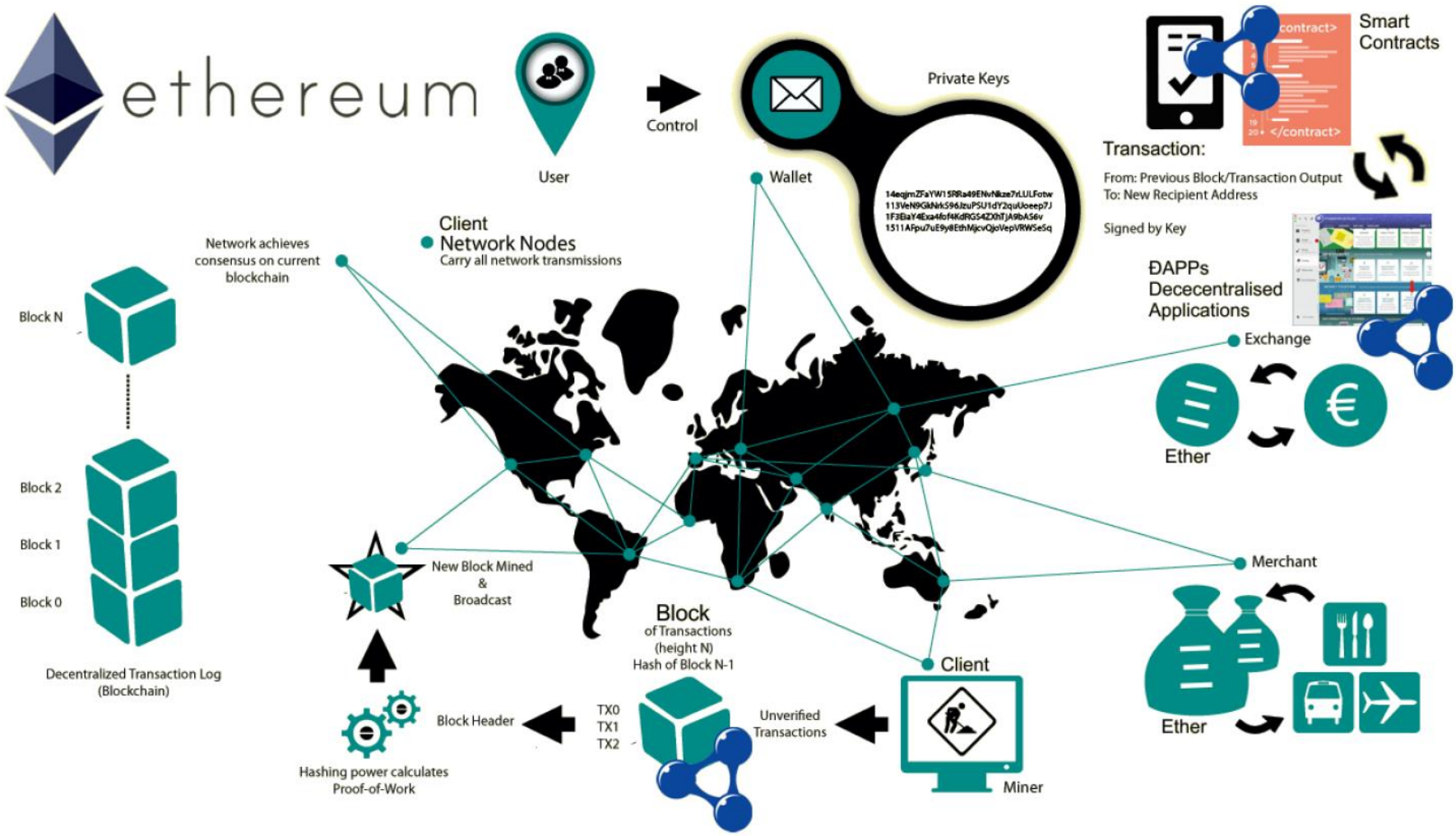


Fig. 6. Ethereum framework elements, modified from [39, p.16]

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

Ethereum DApp Architecture

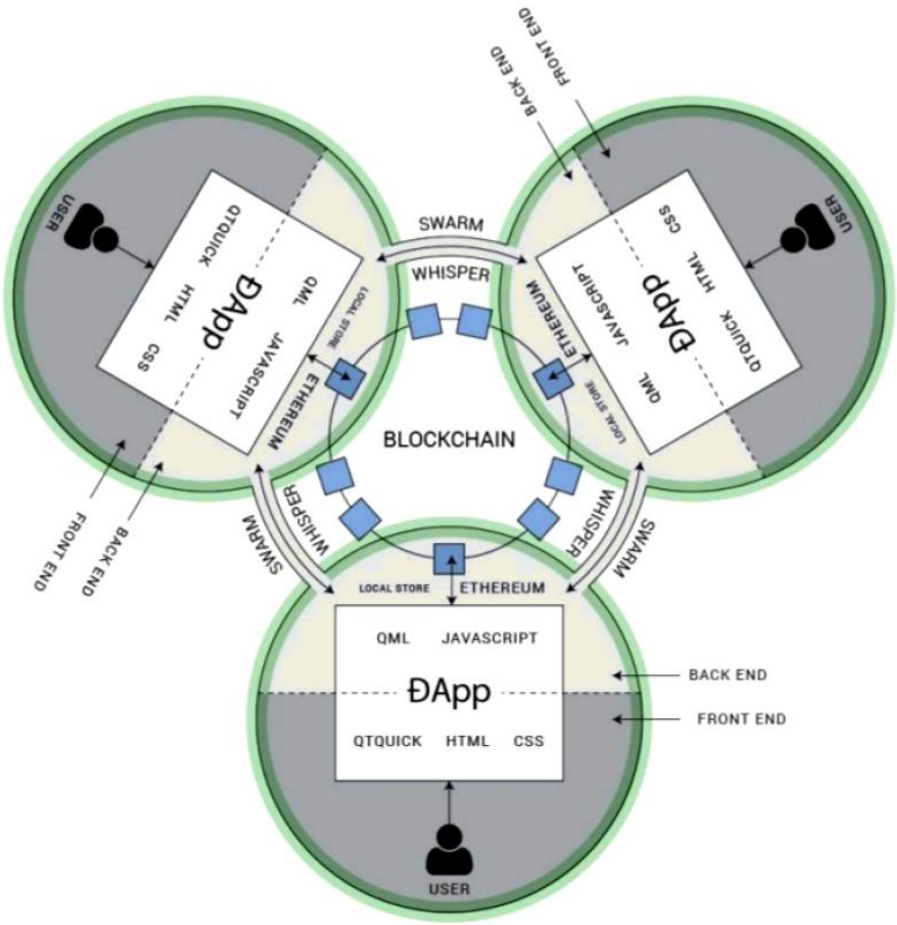
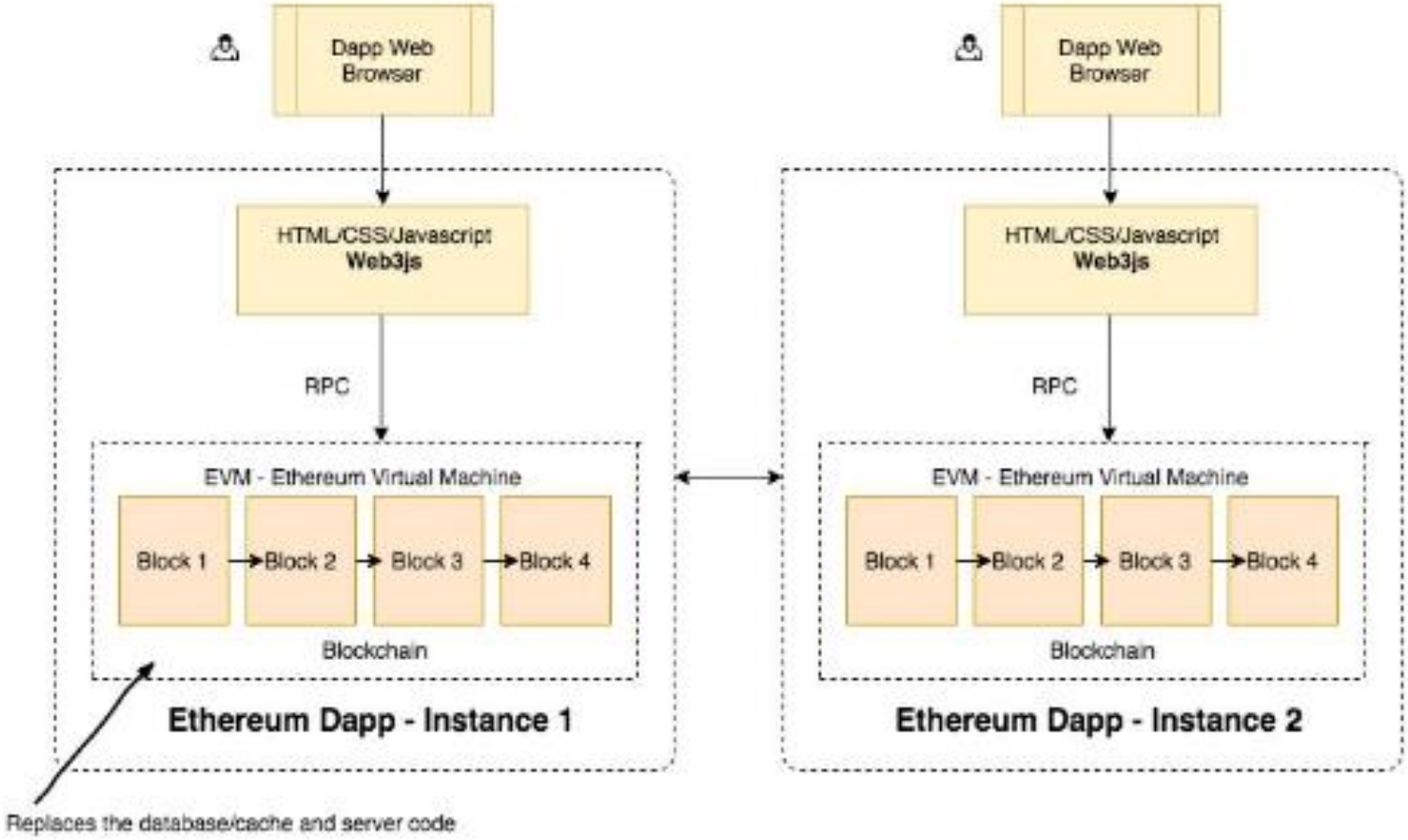


Fig. 11. Ethereum Architecture [52]

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



High-Level DApp Architecture

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

Source: Ethereum Smart Contract Development by Mayukh Mukhopadhyay

Web3.js Tech Stack

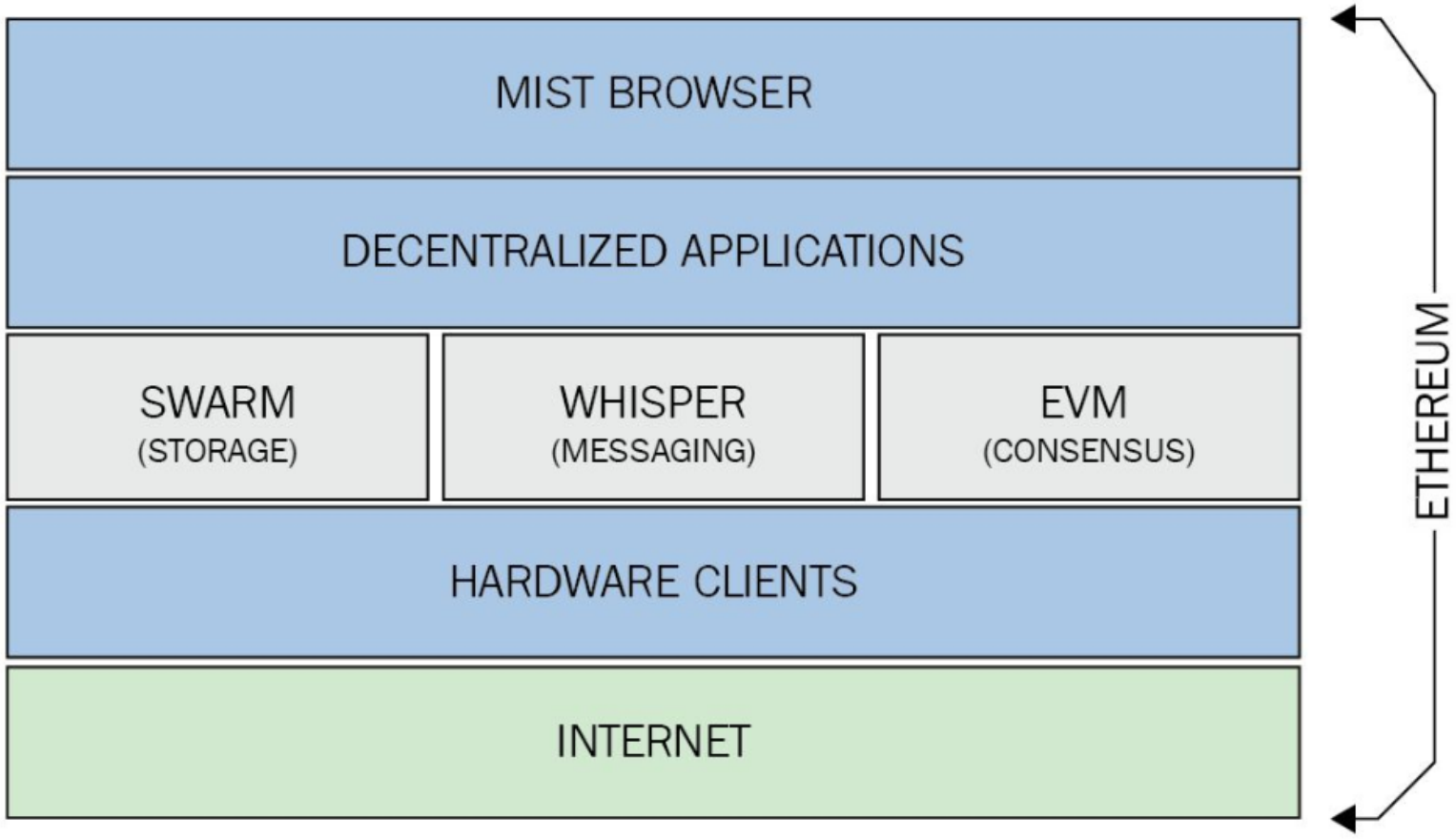
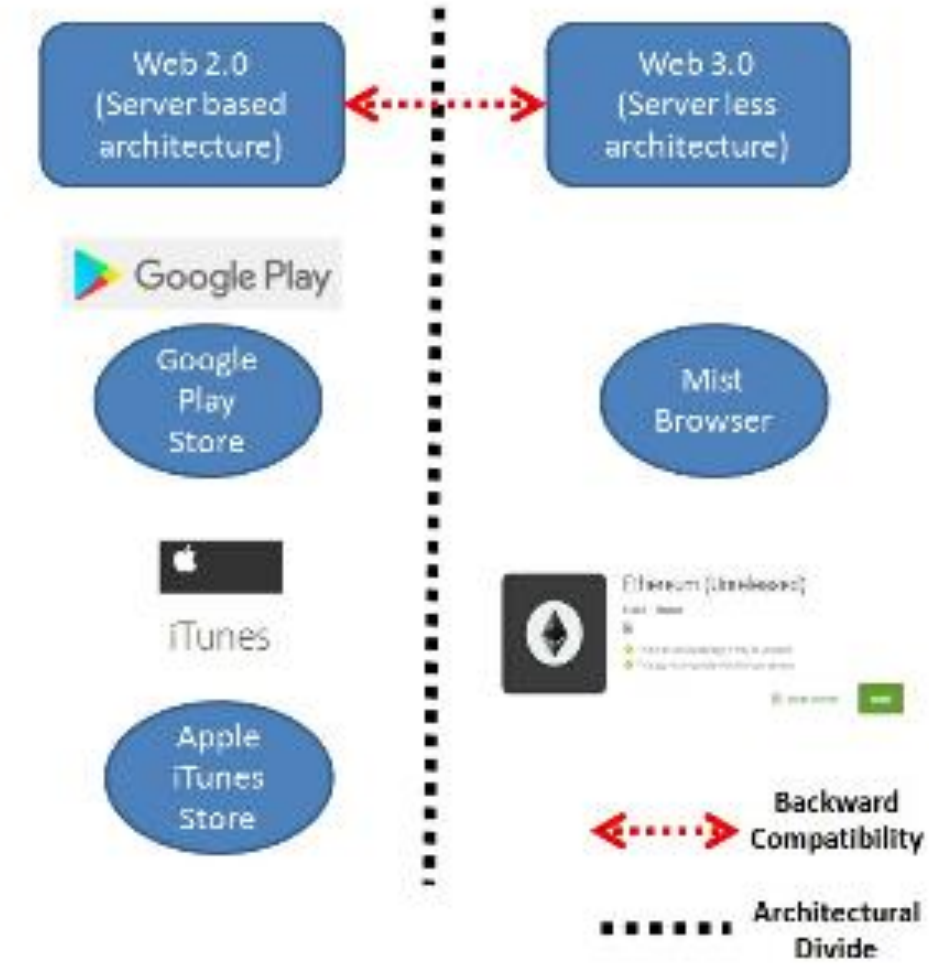


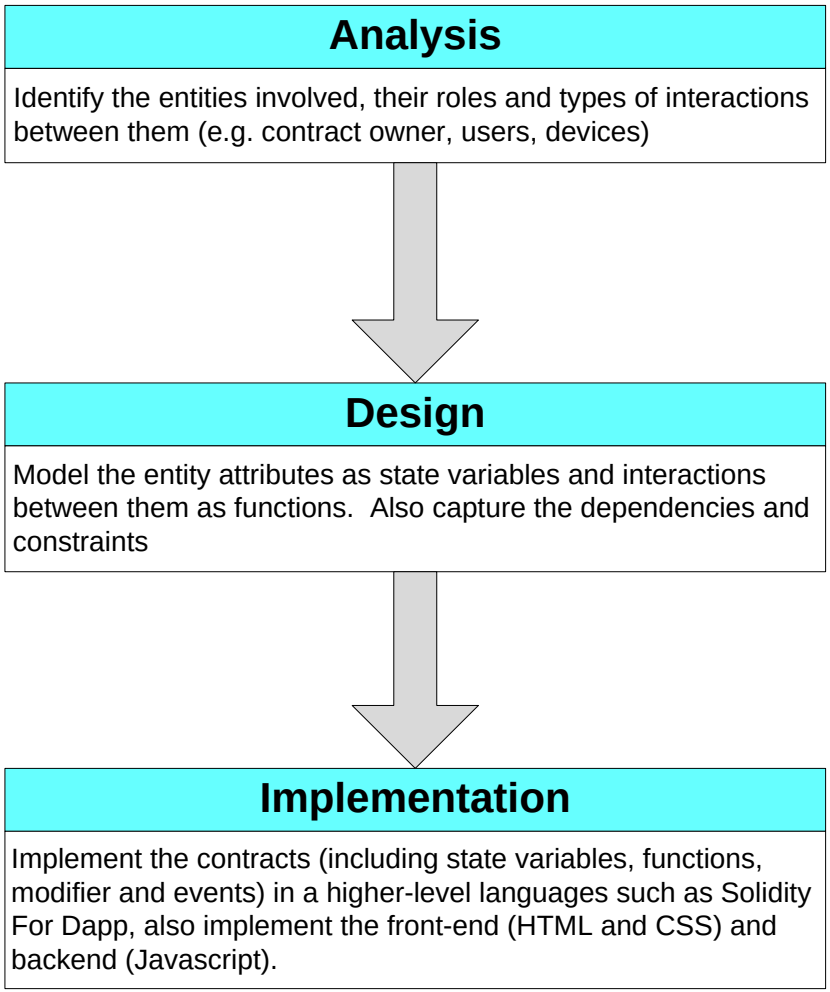
Figure 2.4: Web 3.0 tech stack for Ethereum, Source: Ethereum stack exchange

Web Apps and DApps - Compared



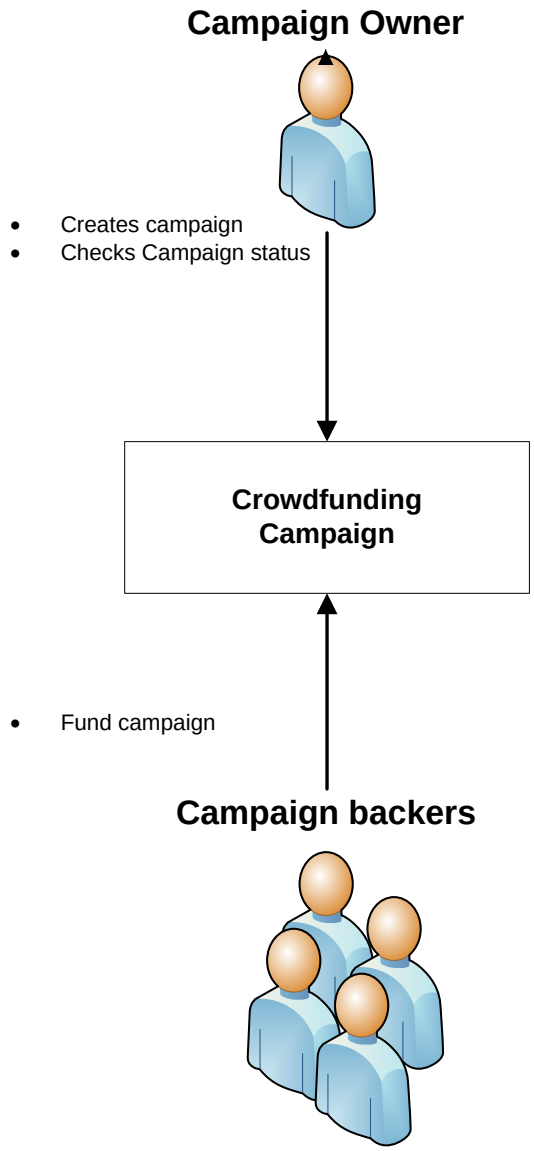
DApp Development Steps

1. Analysis
2. Design
3. Implementation



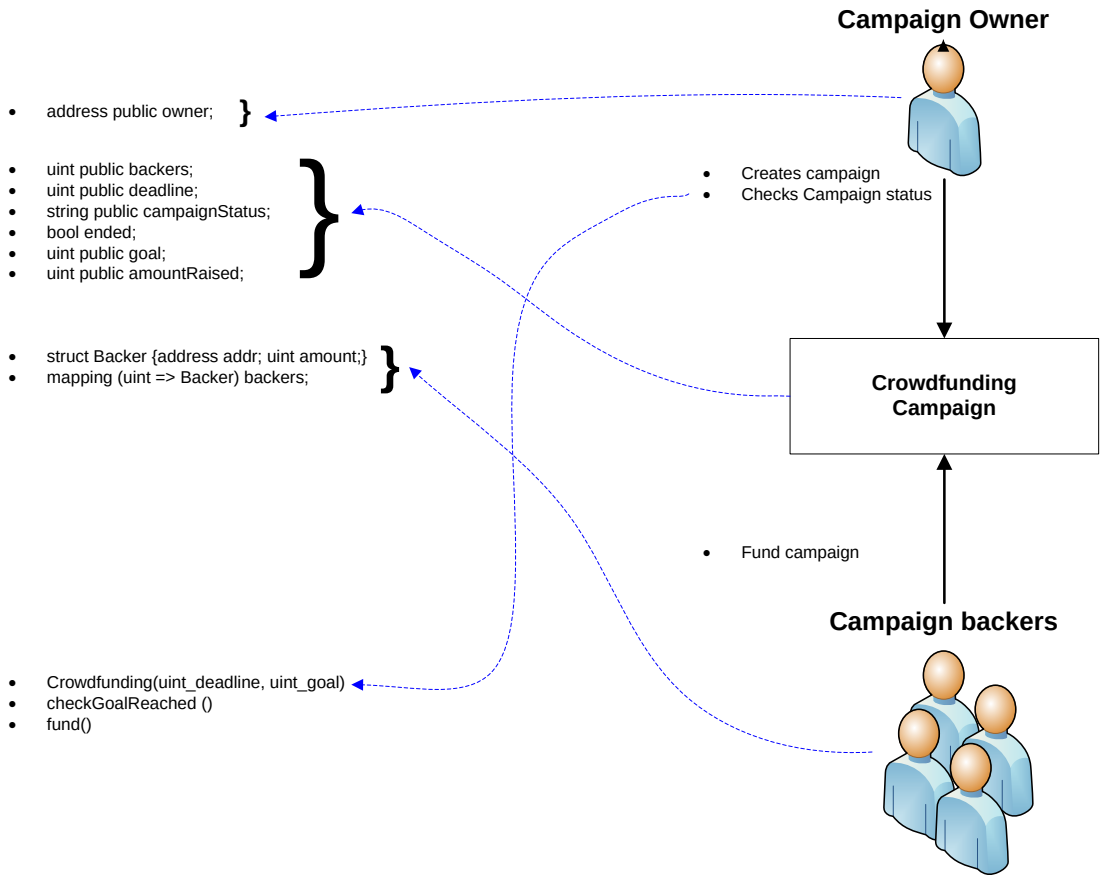
Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti

DApp Development Steps – Analysis - Example



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

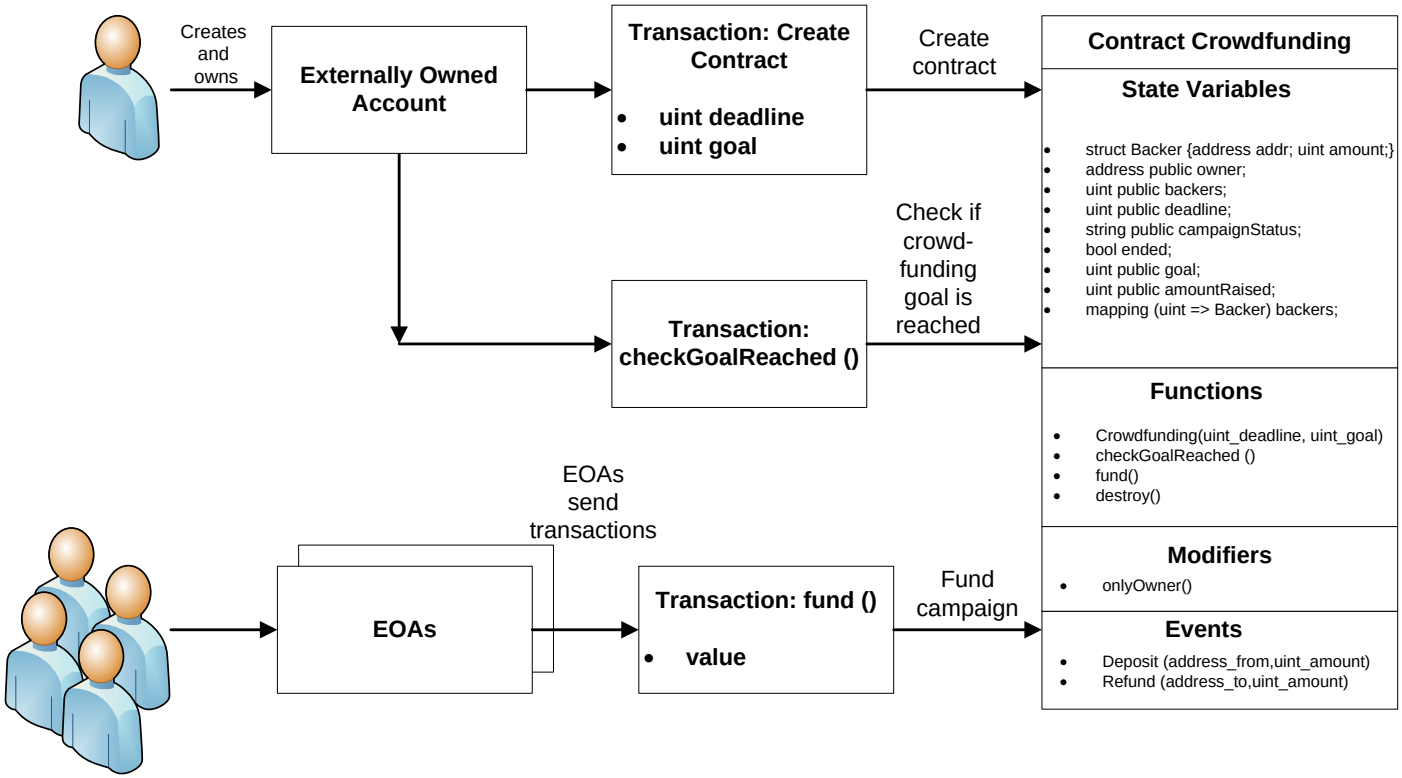
DApp Development Steps – Design - Example



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

DApp Development Steps – Implementation - Example

(Example Business Case: Crowdfunding Application)



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

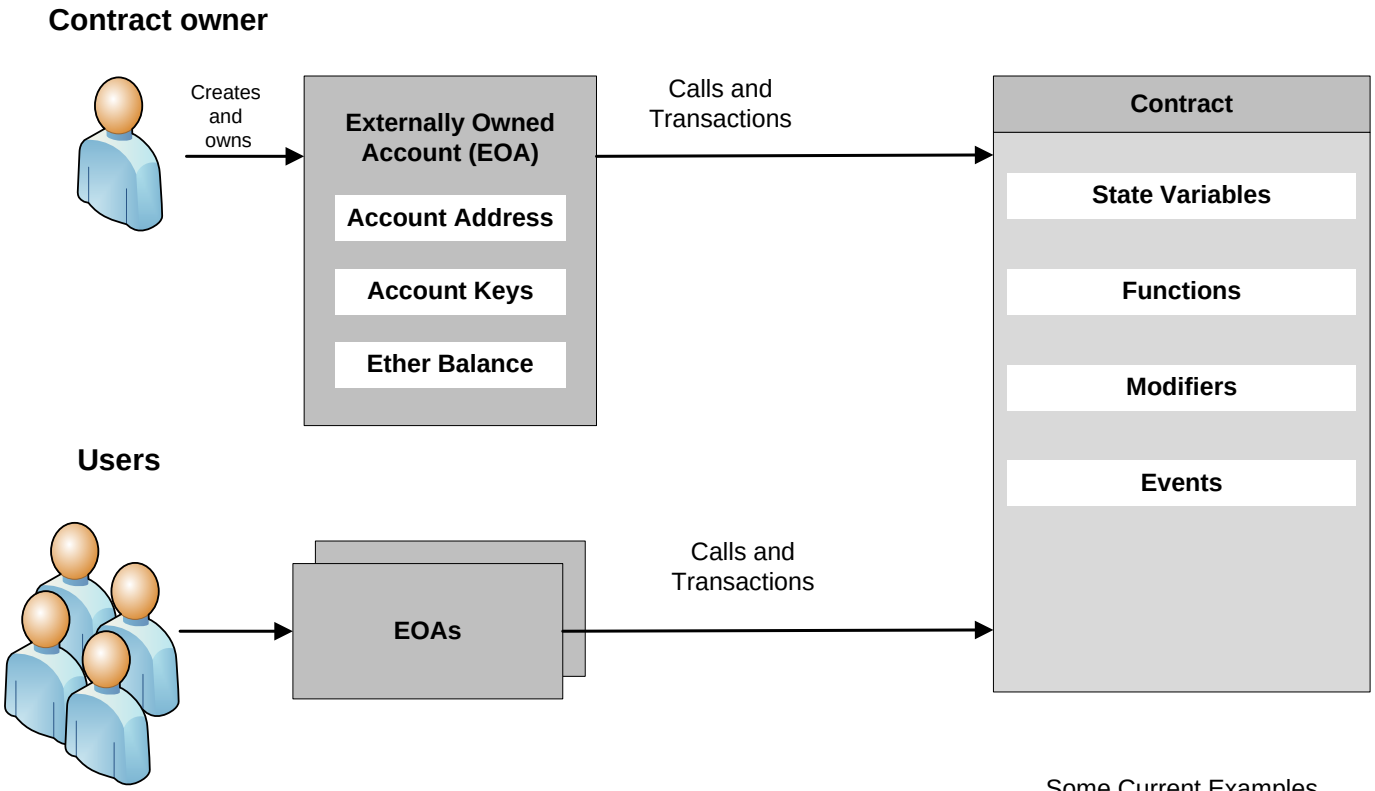
BLOCKCHAIN APPLICATION TEMPLATES

BlockchainWeekly

Powered by Shindig

Blockchain Application Templates

Many-to-One



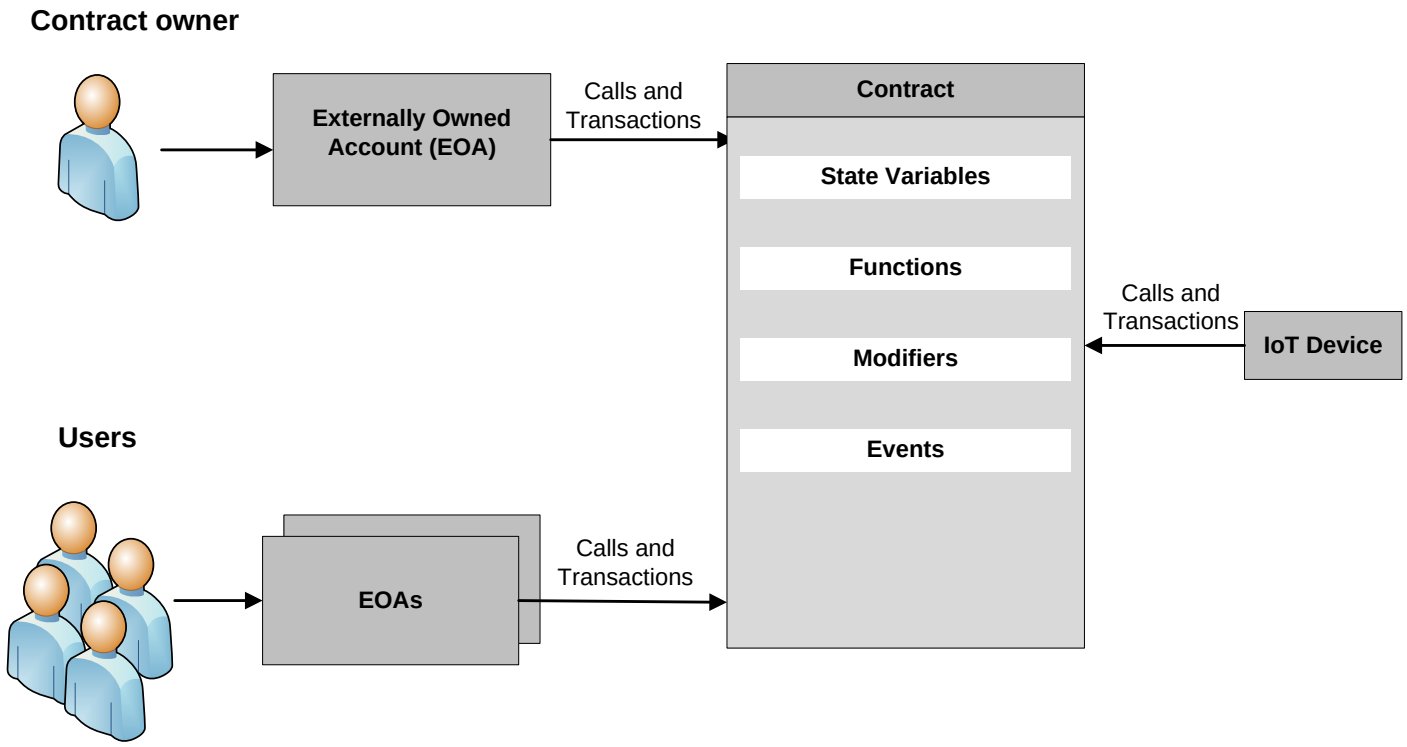
Some Current Examples

- Crowdfunding
- Event Registration
- Voting
- Name Registration

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

Blockchain Application Templates

Many-to-One for IoT Applications



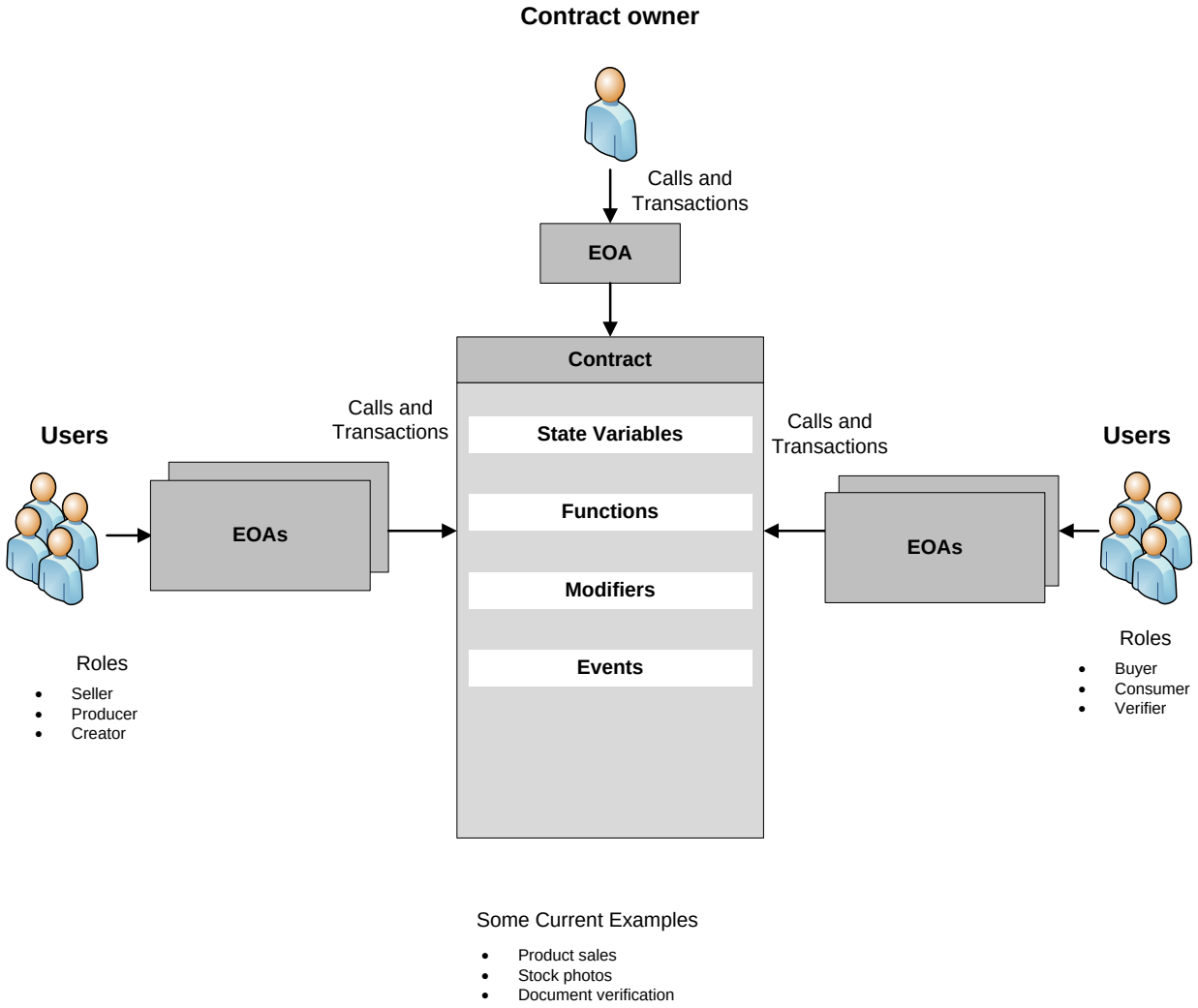
Some Current Examples

- Solar charging stations
- Smart switch

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

Blockchain Application Templates

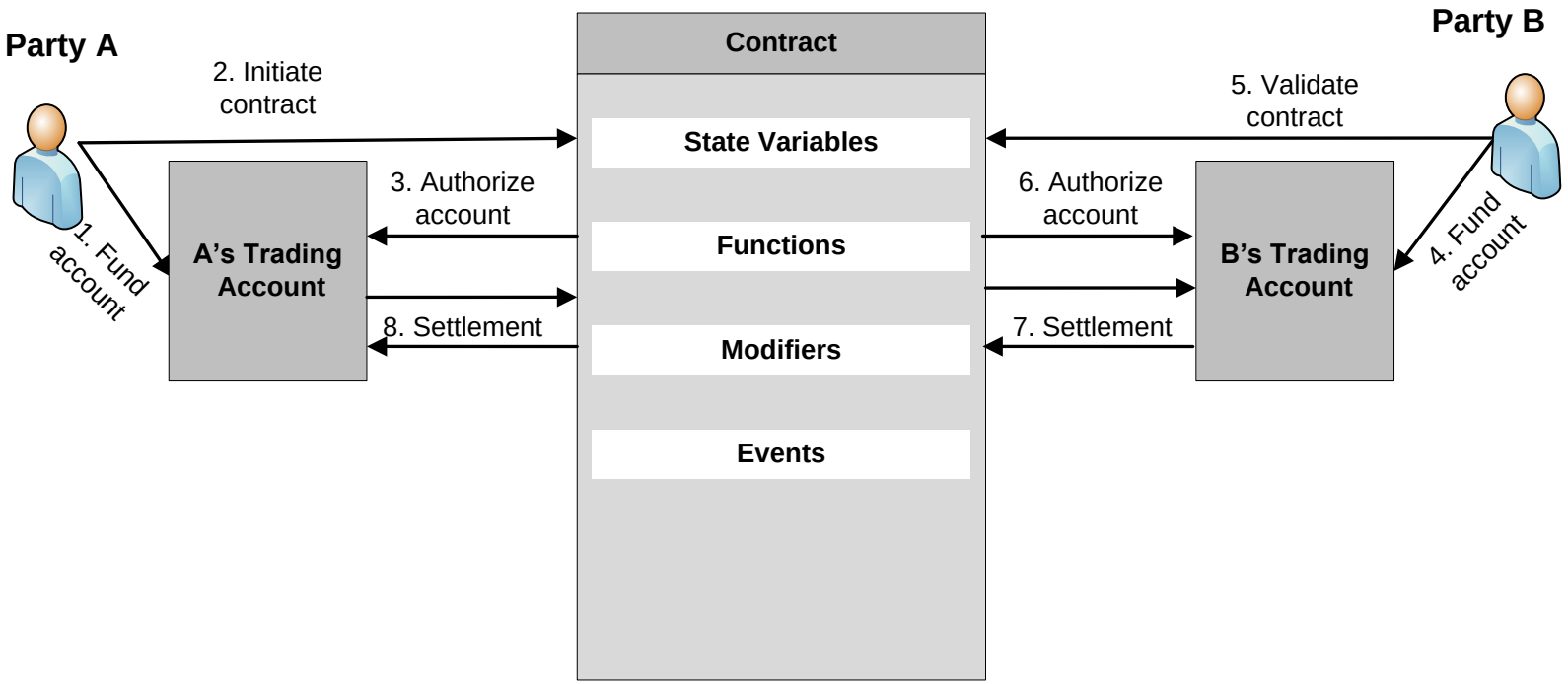
Many-to-One for Financial Applications



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

Blockchain Application Templates

Many-to-Many or Peer-to-Peer

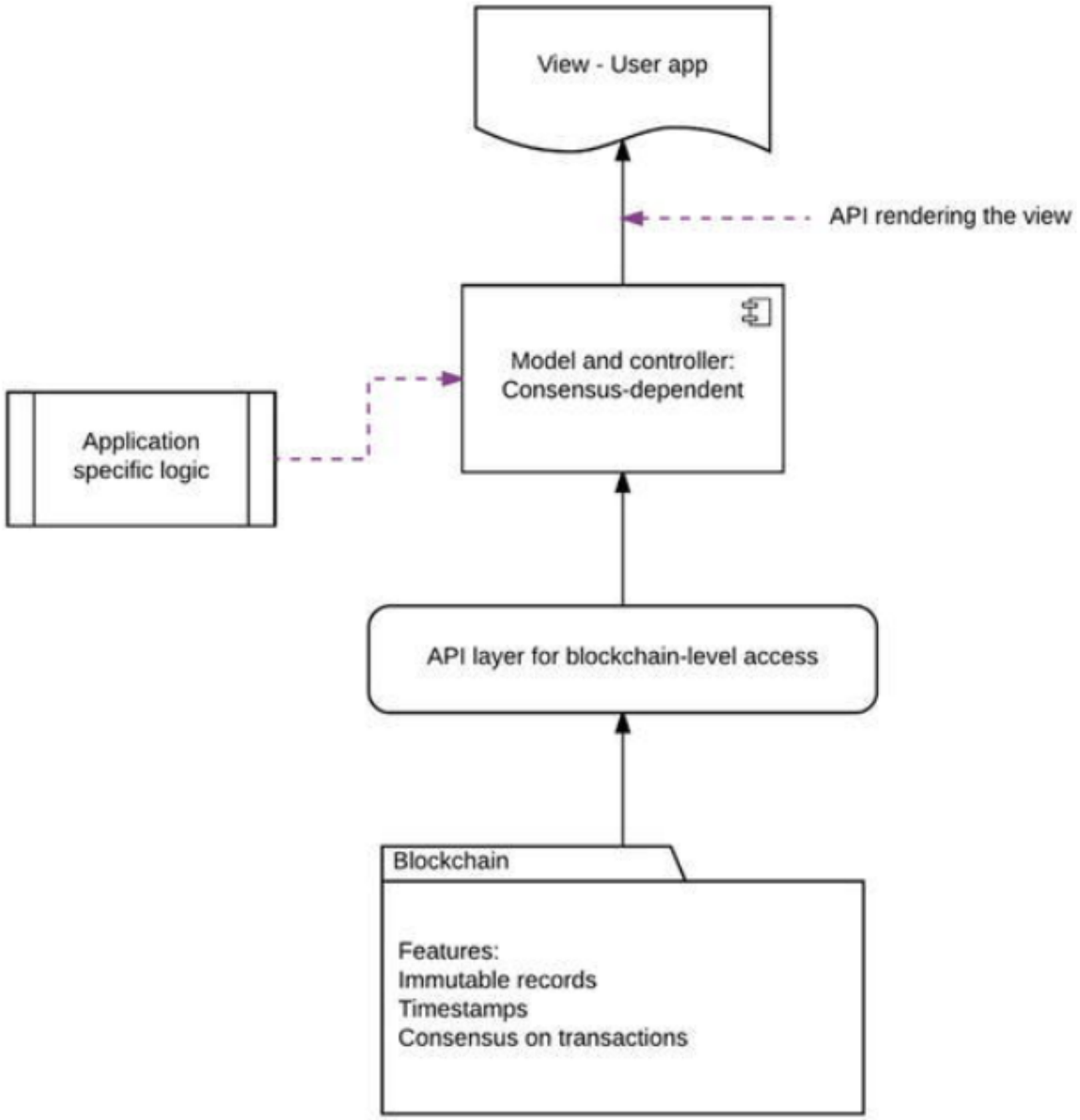


Some Current Examples

- Call option
- Interest rate swap

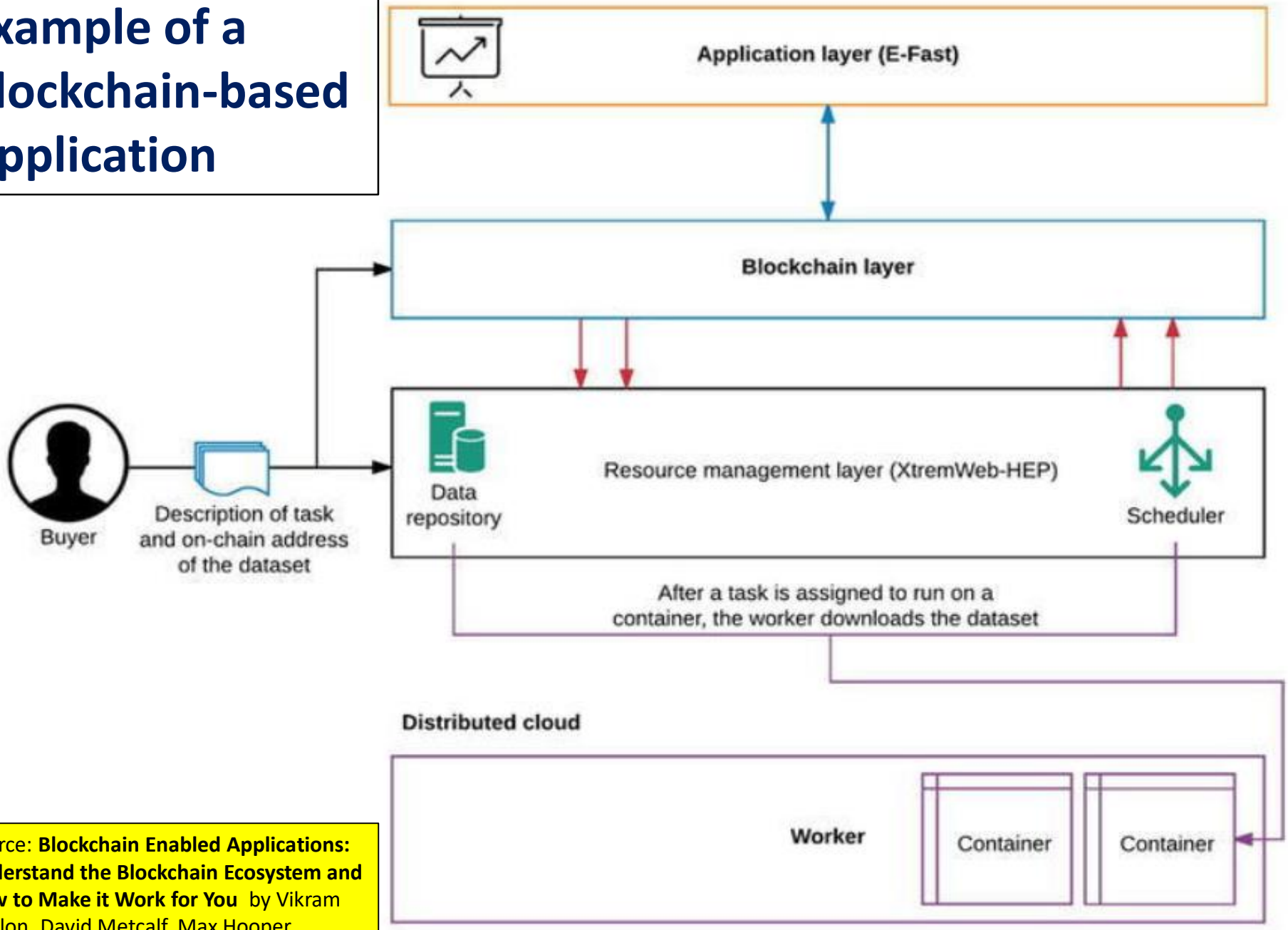
Source: Blockchain Applications: A Hands-on Approach by Arsheep Bahga and Vijay Madisetti

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-based Application



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

TOPIC 9: HOW TO HELP YOUR ORGANIZATION RAPIDLY RAMP UP SKILLS AND READINESS FOR BLOCKCHAIN APPLICATION DEVELOPMENT

BlockchainWeekly

Powered by Shindig

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.

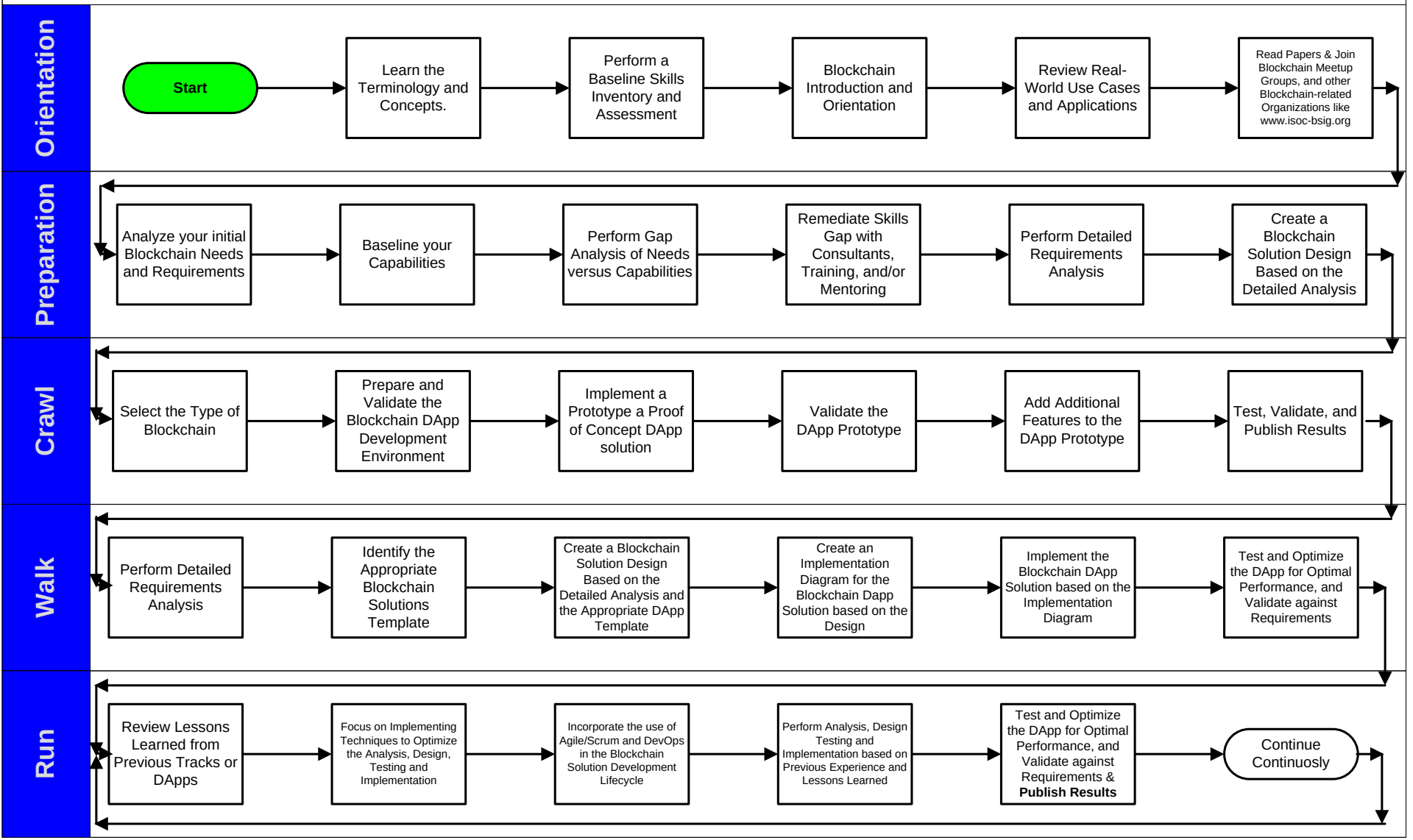
1	397 (100.00%)	Blockchain	15	111 (27.96%)	Smart Contracts
2	200 (50.38%)	Finance	16	107 (26.95%)	Solidity
3	184 (46.35%)	JavaScript	17	106 (26.70%)	Linux
4	168 (42.32%)	Node.js	18	104 (26.20%)	AngularJS
5	151 (38.04%)	Ethereum	19	101 (25.44%)	Docker
6	146 (36.78%)	Bitcoin	20	98 (24.69%)	Redis
7	142 (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 Required Skills for a Blockchain Development Staff

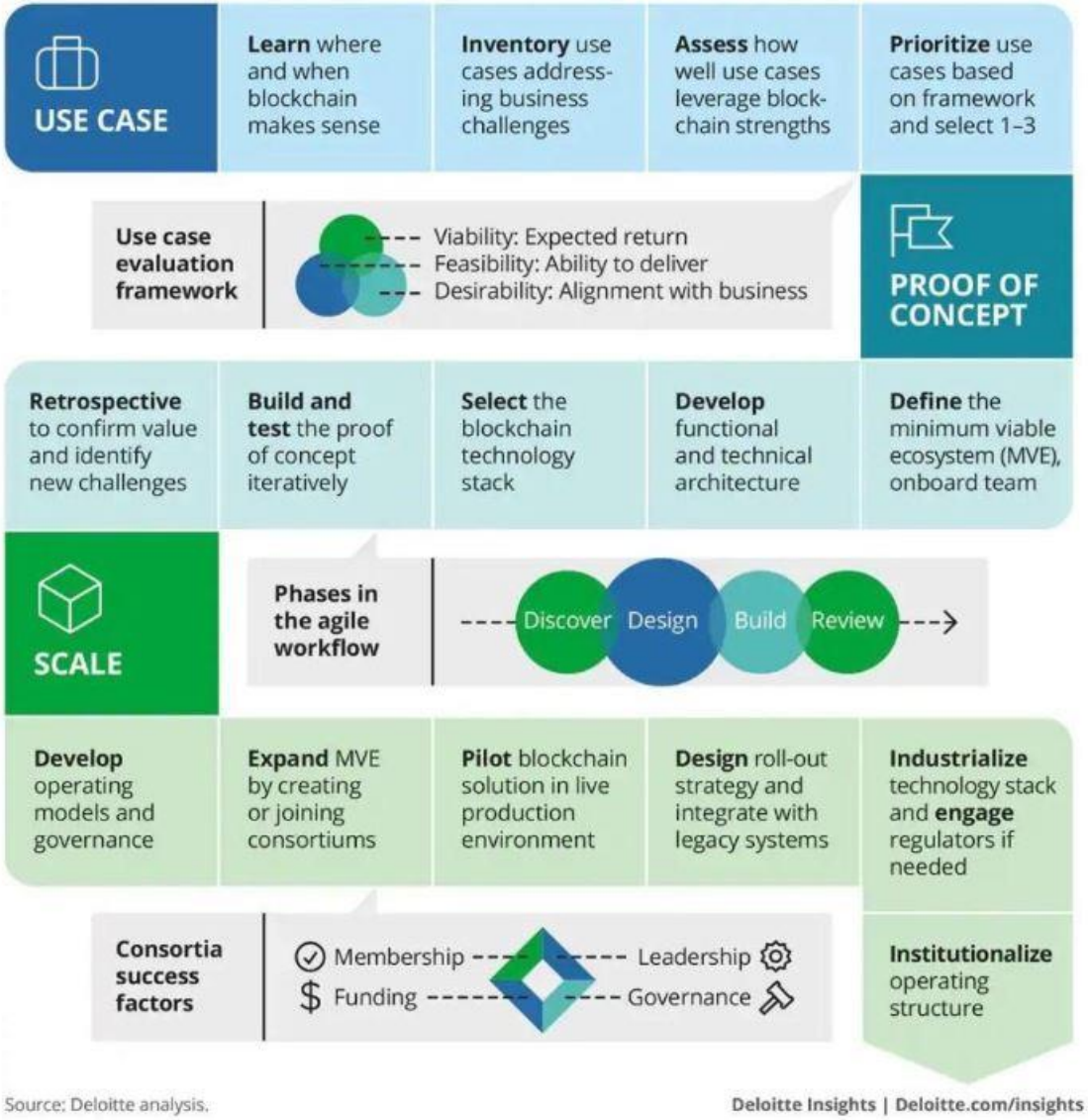
- Web3.js
- DApp development
- UI and UX Design and Testing Skills
- Deep understanding of compiled code, Gas, and the Ethereum Virtual Machine (EVM)
- Secure coding
- Defensive coding
- Egoless Programming
- Stringent Code Reviews
- Networking
- Understanding of Protocols
- Planning
- Requirements
-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

And don't forget PROJECT MANAGEMENT & PROGRAM MANAGEMENT!

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



The Blockchain Implementation Roadmap



CONCLUSION

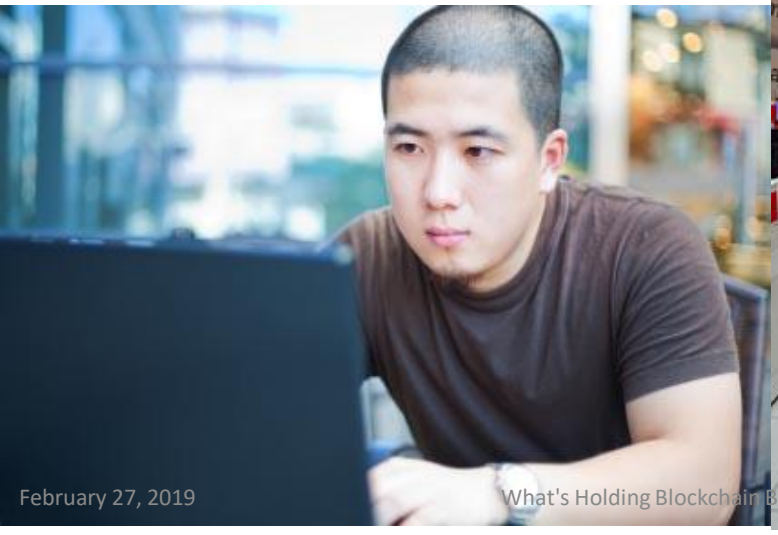
BlockchainWeekly

Powered by Shindig

Conclusion

So we covered:

- Why Blockchain?
- Blockchain Law
- Distributed Systems and Blockchain Security Concepts
- Blockchain Limits and Challenges
- How to Secure Blockchain Infrastructure and Applications
- How to perform Secure Software Development for Blockchain applications by design, coding practices, testing and verification
- Blockchain and Auditing
- 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.

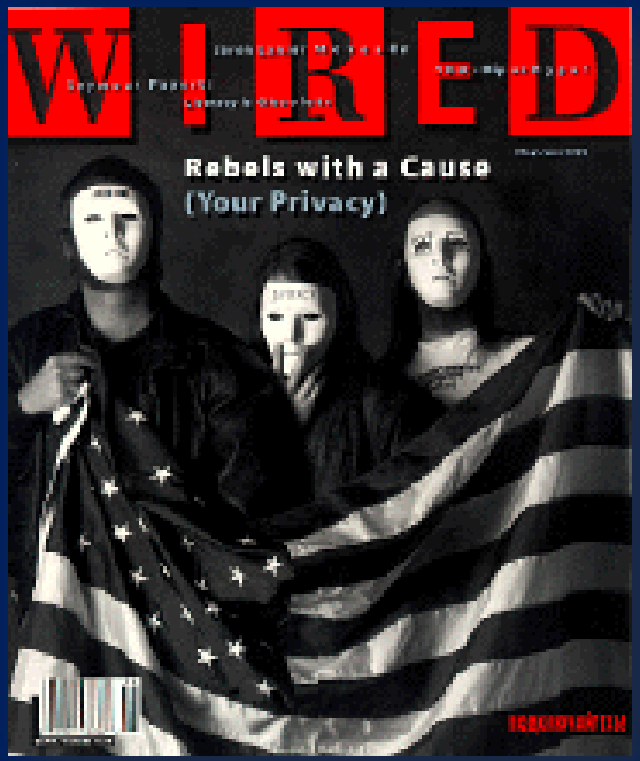
Source: **Blockchain still vulnerable to hacks despite security hype, but here are some solutions by James Nguyen. Retrieved from https://e27.co/blockchain-still-vulnerable-to-hacks-despite-security-hype-but-here-are-some-solutions-20190212/ -**

QUESTIONS

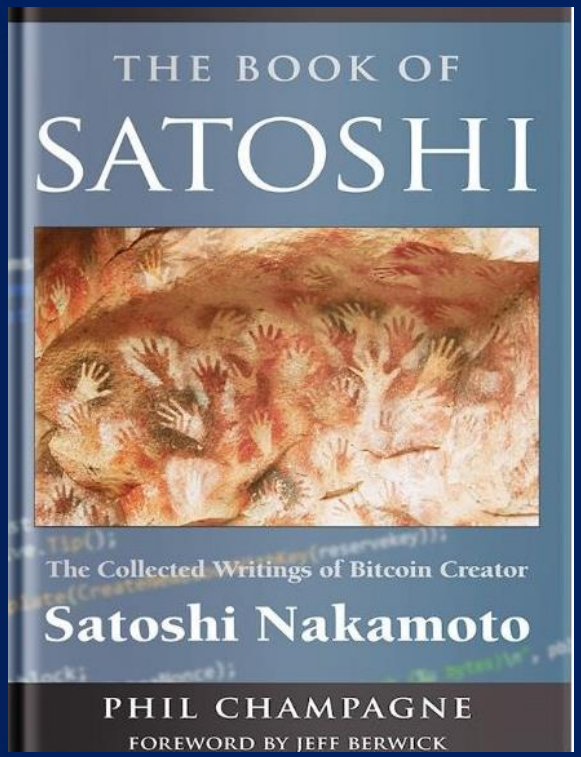
BlockchainWeekly

Powered by Shindig

Questions?



Crypto Rebels
Revealed
Wired Magazine,
February 1993



Book of Satoshi
Collected Writings
Of Satoshi Nakamoto

General George S. Patton

BlockchainWeekly

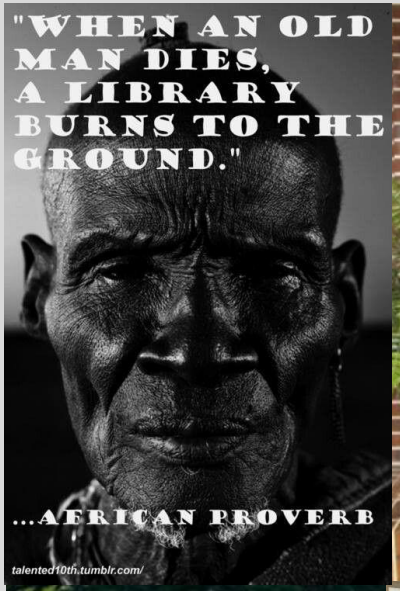
SPECIAL THANKS

BlockchainWeekly

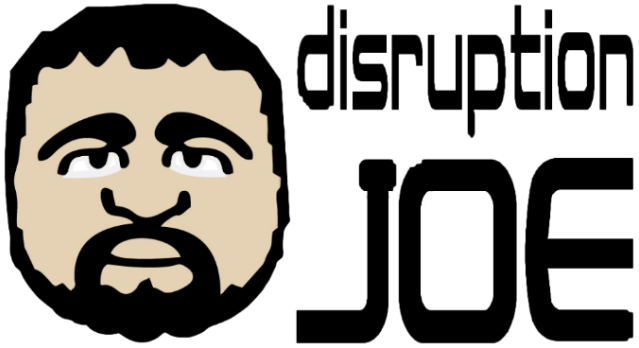
Powered by Shindig

Dedication

- This work is dedicated with love, admiration, gratitude, and great respect to ***James P. Jarnagin*** (January 25, 1935 – December 2, 2018), the man who was my Mentor and Father-figure since March 1985. He is one the biggest reasons for my career success and personal success. What I owe him can never be repaid.



Special Thanks To
Chicago's Best Blockchain Buddies:



Joe Hernandez
Co-Founder of the
Chicago Blockchain Project

Hannah Rosenburg
Co-Founder of the
Chicago Bitcoin and Open
Blockchain Meetup

ILLINOIS INSTITUTE OF TECHNOLOGY

Special Thanks To:

Vitalik Buterin
Inventor of Ethereum
@VitalikButerin on
#Twitter

BlockchainWeekly

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

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 .

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.

References

- 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.
- 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.
- 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 Texts

- **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
- **Foundations of Blockchain**
  - **By Koshik Raj**
- **The Book of Satoshi: The Collected Writings od Bitcoin Creator Satoshi Nakamoto**
  - By Phil Champagne