Blockchains 2018

DigiSoft

digisoft.com.uy
A.N.I.I. (anii.org.uy)
HPI_X_2017_1_114811
2018-08-15

Preface

This review of blockchain technology and its uses was completed in middle 2018 under a five month contract with Digisoft SRL[1], a Uruguayan software consultancy. Digisoft was itself under contract with the research agency ANII (*Agencia Nacional de Investigación e Innovación*[2]) of the government of Uruguay. The funds provided for a review of blockchain technology and for an exploration of potential business applications of the technology[3]. The tremendous potential offered by blockchain technology warrants an in-depth review of the technology and of projects based on it.

This review aimed to take stock of blockchain technology today, sort out the real promise from the ethereal potential, understand the shared experience and discoveries, and establish the usable foundations on which progress can be made. After a historical exploration, this review evaluates blockchain technology at three levels: as a data structure, as a computational project, and as an information system.

This review must be read with some caution. The text is written from an outsider's perspective: the author did not participate in the discussions, development, running, or use of any blockchain project. The text is necessarily incomplete since there has been so much activity in so many different directions. Also the text is already outdated: new ideas, issues, directions, and projects are being developed continuously in the current flurry of innovation.

This review consists of an introduction followed by five parts. The first part summarizes the history of blockchain projects from antecedent developments in computer science and popular computing, through the origin of Bitcoin, the emergence of a second generation able to store computer code and execute it later on demand, to the current emergence of third generation, formally constructed blockchain projects. The second, third, and fourth parts present blockchain technology at various levels: as a data structure, as a distributed computational project, or as information systems. The final part presents the research projects undertaken in complement to this review.

The end of the first decade of a technology provides a good moment for a retrospective review. The first, fully functional, widely used project based on a distributed ledger, *Bitcoin*, was proposed in 2008 and went live in 2009. A decade later, in 2018, we have sufficient perspective and experience to sort technological success out from promises, hopes, or dreams.

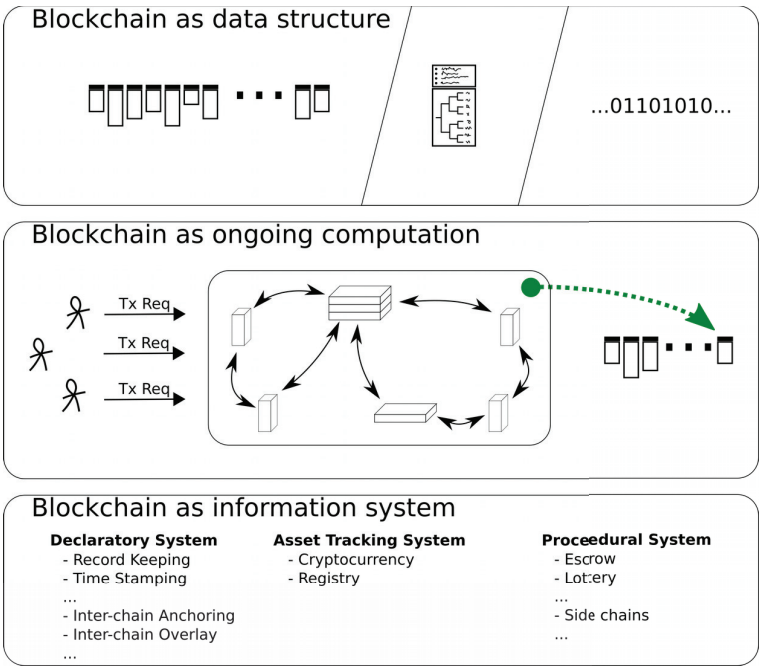
1 Digisoft SRL
2 Agencia Nacional de Investigación e Innovación
3 Contract: `ANII HPI_X_2017_1_144811`

Table of Contents

Introduction

Understanding the world of blockchain information systems, in 2018, is a daunting challenge. One needs to acquire conceptual, technical, historical, and social knowledge about myriad of efforts in a multitude of different directions discussed by a plethora of different participants. This enormous diversity of technologies, projects, and communities makes it difficult for newcomers to understand the field. Unfortunately, in order to contribute with new designs, with software code, with business plans, one must fully understand the domain.

In order to make sense of these efforts, directions, and discussions and in order to organize our own evaluation of them, we structure our thinking by considering the world of blockchains at three different levels:



As a data structure, the self-validating nature of blockchains ensure they form the basis for data storage with archival guarantees. As an ongoing computation, a blockchain project involves people using machines connected into a network running code to serve some purpose under common agreement. As an information system, a blockchain project permits specific business activities such as time stamping, cryptocurrency based payment systems, digital registries, escrow services, and even cross-linking separate blockchains.

Blockchain as Data Structure

Blockchains, viewed as data structures, are relatively trivial, with a structure akin to a linked list. Blockchains act as extensible, append only, data stores with cryptographically secured integrity guarantees.

The data structures used for blockchains depend heavily on cryptographic hash functions, which are functions which calculate a fixed size value (a hash function) with specific properties such as the

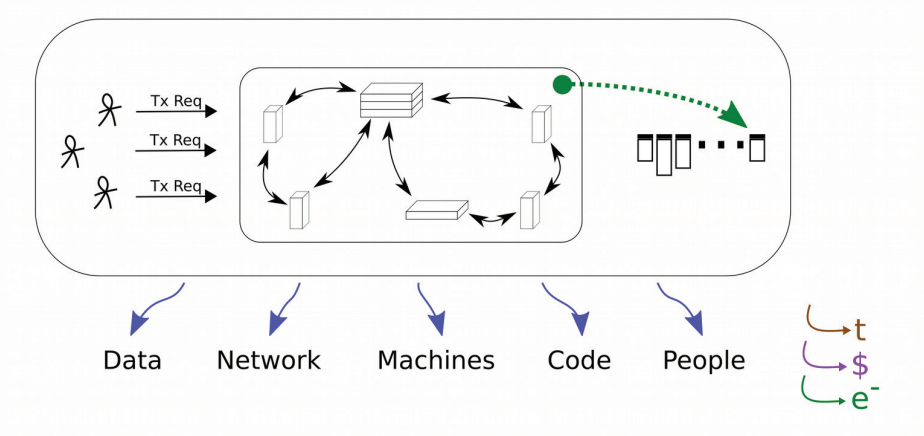
inability to calculate the input value from the output hash value. These properties depend on our current knowledge of mathematics.

Blockchain data structures, at their most basic, involve a series of binary blocks with each block after the first including somewhere within itself the value of a cryptographic hash calculated from the proceeding block. This necessarily requires defining which cryptographic hash is used and where the value is stored. In general, for questions of computational efficiency, the blocks are structured internally into a header and a body, each of which has its own internal structure. When the body of a block contains multiple elements, these are generally structured into trees. The information must be serialized into a binary format suitable for streaming or storage.

Blockchain data structures offer tremendous potential for information systems. When used individually, they serve as the foundation of archival data structures. When exchanged between parties, blockchains can serve as distributed data bases. Blockchains already have many uses. The modern filesystem ZFS uses hashing much like a blockchain to store data and ensure its integrity. The distributed version control systems Mercurial and Git use what are essentially blockchains to store and track changes in sets of files, mostly for development of software source code. Cryptocurrency systems like Bitcoin and Ethereum use blockchains as a distributed, shared, and public ledger of all past transactions to produce and exchange provably scarce informatic resources considered to be tokens of the cryptocurrency.

Blockchain as Computation

Blockchains, viewed as ongoing computations, involve a collective computing project using a blockchain data structure as the shared, replicated, append only database for recording the history of the computation.



Blockchain computation projects involve people using machines connected into a network running code to serve some purpose under common agreement. The involvement of people implies an investment of time, money, and energy in the project. The machines apply processing power, dynamic memory, and storage to the computation. The necessity for a network implies the definition of messages, protocols for their exchange, and a communication network for their transport. The code running on the machines includes code to process the data, client and server code for networking, and code to query the system; all of these require design, development, and maintenance. The necessity for synchrony of the system requires common agreements on goals,

procedures, discussions and decision making, commonly referred jointly as self-governance of the blockchain project's community.

Many analytic approaches can be taken to examine these computation projects. The history shared by the participants offers one approach. Another approach is to consider these systems as autonomous legal jurisdictions. A financial analysis could consider these projects as economic systems and examine the incentives for and expenses of its participants. An ecological approach could consider the environmental impact of the effort. Here, we focus on the information system implicit in these projects.

Blockchain as Information System

Blockchains can be viewed in the abstract, as information systems. These involve the application of a blockchain computing project to a specific business need.

Conceptually, blockchains can be viewed as information systems of four major types. Blockchains can act as *declarative systems* which record and archive data. These systems could serve to log data from disparate sources into a single blockchain archive. Alternatively, such systems could allow users to timestamp the occurrence of an event or the existence of a digital resource. For example, it might be possible to setup a blockchain system to track a complete electoral process into an authoritative archive of the vote. Blockchains can also act as *asset tracking systems* determining the rules through which to define an asset, to generate the asset or instances of the asset, to establish the ownership of the asset, to transfer the asset to another owner, and, possibly, to withdraw assets from the system. Such asset tracking systems can be used for virtual, on-chain assets such as digital currencies or to act as formal registries for real-world assets such as land or real-estate. The second generation of blockchain systems gained the ability to store computer code which could be invoked and executed subsequently. These blockchains can act as *stored procedure systems* able to perform much more complex information systems. The stored procedures, so-called `smart contracts,' can provide single services such as an escrow service or a lottery. Several blockchain projects can be tied together to act as a *multi-blockchain system*. The information system might include formal connections between blockchains such as one, low usage blockchain anchoring itself to a more popular blockchain thereby gaining characteristics of the latter. Current research is actively seeking the ability to move tracked assets between blockchain systems, withdrawing the assets from circulation in one blockchain while simultaneously releasing the assets for circulation in the second.

The next chapter will examine the history of blockchains from antecedents through the first generation blockchain project, Bitcoin, the second generation blockchain project, Ethereum, and on to the current generation of efforts.

Blockchain History

This historical review of the world of blockchain distinguishes four historical eras

separated by three historical events, the release of Bitcoin in 2009, the release of Ethereum (Classic) in 2015, and the release of the new Ethereum in 2016, although the later date is analytically convenient rather than historically significant.

The antecedent historical era for blockchain projects covers the period from the start of the digital revolution following the Second World War to the release of the first blockchain project, *Bitcoin*. This era provided the necessary cultural and technological precursors from which Bitcoin could be invented, including the rise of general purpose computing, the emergence of widespread network connectivity, the invention of modern digital cryptographic primitives, and the rise of a pool of competent programmers, the tools for their collaboration, and shared desire to develop solutions focused on the needs of the individual.

The first historical era of blockchain projects started with the release of the *Bitcoin* blockchain project at the start of 2009 and ended with the release of the second generation blockchain project, *Ethereum*, in the middle of 2015. Retrospectively this can be called the *Blockchain 1.0* era, in which the community discovered the basic functionalities of blockchain systems. The release of Bitcoin demonstrated that a solution was possible for cryptocurrencies, the decentralized generation and exchange of scarce informatic resources. In this era, the discussion focused on how to maintain a blockchain project, how to improve the system, and how to apply the system to new uses. These first years of the Bitcoin project included its initial rise in popularity, principally due to its use for payment for illicit substances on the deep web marketplaces.

The second historical era of blockchain projects started in the middle of 2015 with the release of the first of the second generation blockchain projects, *Ethereum*, which allowed users to embedded software code in the blockchain archive and invoke that code later, a massive expansion in capabilities of blockchain systems. The era can be considered to end, for convenience rather than any historical precision, with the fork of the new *Ethereum* from *Ethereum Classic* in July 2016, following the debacle of the massive shared cryptocurrency fund called *The DAO*. This era can, retrospectively, be called the *Blockchain 2.0* era, in which the community discovered the basic functionalities of fully automated stored procedures, the so called *smart contracts* or *chaincode*. In these years, the rise of popularity in blockchain projects came from the exponential rise in value of the tokens used as cryptocurrencies and from the myriad of new uses made possible by the technology, ranging from voting, through land registries, to supply-chain management. The era ends with the realization of the drawbacks of relying on computer code as the sole arbiter of exchange and with the acceptance that multiple blockchain projects will co-exist due to having different focuses.

The third historical era of blockchain projects started with the split of the first of the second generation blockchain projects and continues to today. The era can be called the *Blockchain 3.0* era, in which rising interest leads to an explosion of projects but the community is tempered by a realistic view of the costs and limitations of blockchain projects. In this era, the first and second generation blockchain projects continued working but the focus of developer interest and

excitement expanded in several new directions. One direction involved building third generation blockchain projects, such as Cardano, focused on robustness, both of design and code, and on solving the issues which plagued earlier projects, such as scaling and self-governance. Another direction of innovation has come from the efforts to provide software tools from which custom blockchain projects could be built, such as the projects in the Hyperledger foundation. A third direction has been the new work to leverage stored procedures (smart contracts) to provide services to the blockchain and to external applications. The popularity of blockchain efforts in this era is tempered by the end of exponential increases in value of cryptocurrencies and the increased suspicion of market manipulation in those values.

Antecedent Era (1945-2009)

The historical era between the end of the Second World War in 1945 and the release, in 2009, of the first distributed blockchain project, Bitcoin, shaped that project directly with influences which still impact the world of blockchains today. The era saw the digital revolution giving rise to personal, general purpose computing, experienced the origin and spread of a global computer communications network centered on the Internet, and witnessed the invention of modern *public key* cryptography. The era also experienced ongoing concentration of political and economic power along with ongoing interventions by governments to control the financial system and repress the emergence of technology outside government control. The era witnessed the formation of digital communities with anarchist or libertarian tendencies able to self-organize, despite being geographically distributed around the world, into collectives who could discuss, design, and develop tools for their own purposes.

Technology

The historical era prior to Bitcoin gave rise to the technological elements which would eventually used by blockchain projects.

The digital revolution of the second half of the twentieth century laid the technological foundations for blockchains. The invention of the transistor, its use within integrated circuits, and then in very large scale integration (VLSI) chips led to Moore's ruthless law[1] which saw computer hardware shrink from buildings to pockets. The reduction in size was paralleled by a reduction in cost, making computers accessible to larger and larger portions of society.

The networking revolution of the later part of the twentieth century had similar effects. Proprietary networking technologies fell aside under the vast expansion of the Internet from its roots in ARPANET (1969), its conversion to TCP/IP (1983), its opening to the public at large through commercial internet service providers (around 1990), and its mass adoption following the "Internet Tidal Wave" memo at Microsoft (1995)[2]. This technological expansion made communication accessible to more and more of society.

The democratization of computing and networking facilitated the rise of digital collaborations. Distributed collaboration arose with the growth of computers, first through mail and print, then over the networks. Open discussion was facilitated by the Internet, through electronic mail starting after 1970 and leading to open access mailing lists, through USENET fora started in 1980's, and through web fora enabled by the World Wide Web after 1990. The distributed collaboration which produced

1 Wikipedia Moore's law Retrieved 2018-07-26
2 Wired May 26, 1995: Gates, Microsoft jump on 'Internet Tidal Wave' 2010-05-26

UNIX led to online collaboration starting with the BSD family after 1977, continuing with the massive GNU project started in 1983, and the Linux kernel project after 1991. These projects, and others, showed that large scale, multi-year collaborations were possible and effective. These projects also established the canonical tools needed for a project: a web site for documentation and to provide downloads, a version control system for computer code, a forum for discussions such as an email list, and, usually, a live communication system such as a channel on an Internet Relay Chat server. The rise of collaborative networking projects, including the person to person file sharing systems Napster in 1999 and BitTorrent in 2001, showed that independent, function specific, network communities could thrive online. Distributed blockchain systems would require these collaborative tools for community formation, discussion, and collaboration.

The invention and improvements of algorithms would provide the building blocks for the design of blockchain systems[1], including cryptographic hashes to create immutable data structures, public key cryptography to establish identity and control, puzzles to produce digitally scarce resources, and synchronization mechanisms to reach consensus between machines. The concept of cryptographic hashes emerged from simple hashing for linked lists developed by H.P. Luhn at IBM in 1953, through Cyclic Redundancy Checks developed by W.W. Peterson and D.T. Brown in 1961[2], to cryptographic hash functions proposed by M. Rabin in 1978[3] and R.C. Merkle in 1979[4], and leading to the development of the MD family of algorithms starting with the MD2 algorithm of 1988 by Ronald Rivest published in RFC 1115 in 1992, the SHA family of algorithms developed by the United States National Security Agency, and the RIPEMD family of algorithms developed at the COSIC research group of the Katholieke Universiteit Leuven. The use of cryptographic hash functions for the storage of large amounts of information in such a way that they could be quickly verifiable and modifiable came from the invention of hash trees or merkle trees[5], as pioneered by R.C. Merkle in 1979[6]. The concepts of public key cryptography emerged, to the public, with the invention of Diffie-Hellman key exchange in 1976, of RSA (1977) and elliptic-curve (1980's) key generation and use for digital encryption and digital signatures. The concept of digitally scarce resources generated by hash puzzles originated with the work by Cyntia Dwork and Moni Naor in 1993 on fighting email spam[7]. This concept would come to be known as *proof-of-work*[8]. The concept of consensus in distributed networks of machines was crystallized in the early 1980's by Leslie Lamport and others; this concept has become known as the Byzantine General's Problem[9]. Various solutions have been proposed to this problem; Bitcoin would solve it with its Proof-of-Work system, with a slow (10 minute) solution time, an evolving level of difficulty, and a longest-chain rule.

---

1 Arvind Narayanan and Jeremy Clark *Bitcoin's Academic Pedigree: The concept of cryptocurrencies is built from forgotten ideas in research literature* ACMQueue 2017

2 W. W. Peterson and D. T. Brown, "Cyclic Codes for Error Detection," in *Proceedings of the IRE*, 49(1) pp.228-235, 1961.

3 M. Rabin Digitalized Signatures," in R.A. DeMillo, D.P. Dobkin, A.K. Jones, and R.J. Lipton (eds) *Foundations of Secure Computing* workshop proceedings Georgia Institute of Technology pp155-167,1978.

4 R.C. Merkle *Secrecy, Authentication, and Public Key Systems* Information Systems Laboratory Technical Report No. 1979-1, 1979

5 Wikipedia Merkle tree Retrieved 2018-06-15

6 R.C. Merkle "A Digital Signature Based on a conventional Encryption Function" in C. Pomerance (ed) *Advances in Cryptology-CRYPTO '87*, LNCS 293, pp369-378, 1988.

7 Cynthia Dwork and Moni Naor "Pricing via Processing, or Combating Junk Mail" CRYPTO'92: Lecture Notes in Computer Science 740 Springer pp 139-147.

8 Wikipedia Proof-of-Work system Retrieved 2018-06-28

9 L. Lamport, R. Shostak, M. Pease "The Byzantine Generals Problem" *ACM Transactions on Programming Languages and Systems* 4(3):387-389 1982

Distributed blockchain systems would require all of these: computers, networks, collaboration, and algorithms. Participants would require computers and networks to be able to set up the blockchain network. Blockchain projects would require the tools of collaboration developed by collaborations like those of free software projects. The blockchain systems would require a selection of algorithms elaborated in the previous half century.

Experience

The experience of the community interested in the peer-to-peer exchange of digital resources prior to the release of Bitcoin involved continuous strife with governments. People were harassed, threatened with arrest or arrested; people's money was seized. These experiences led to a fundamental distrust of government and strong desire to operate beyond its control.

The experience of individuals in financial crises gave rise to a general distrust of government custody of monetary systems. The arbitrary control of currency movements was a fixture of most government systems until recently. Governments have placed restrictions on the ownership of valuables such as gold (such as the US Gold Reserve Act of 1934[1]) and on the exchange of currency (such as in France, ending only in 1989[2]). Furthermore, governments artificially maintain the value of their national currency and sometimes devalue that currency massively at once. Governments can seize money from users directly through actions such as 'civil forfeiture' actions in the United States with no moral justification. The various financial crises, notably the crisis of 2007-2008, further led to core distrust between individuals and governments.

The experience of individuals during the digital revolution gave rise to a general distrust of government custody of technology. The development of computer technology always included participants with anarchist or libertarian inclinations. The phone phreaking movement arose in the 1960s and 1970s to explore and exploit the telephone telecommunication network[3]. The emergence of public key cryptography in the 1970s[4] was followed by the Cypherpunk movement in the 1980s[5]. The hacker culture[6] of the 1960s and 1970s gave way to the Free Software movement in the 1980s[7]. The emergence of the Internet as the dominant global network led first to the foundation of the Electronic Frontier Foundation[8] aiming to protect individual civil liberties on the Internet[9] and then to John Perry Barlow's "A Declaration of the Independence of Cyberspace"[10] aiming to separate online communities from government control[11].

The experiences of academics during the discovery and initial discussion of public key cryptography and of programmers during its early implementation in software gave rise to a general distrust of government control of knowledge. This conflict came to be known as the Crypto wars[12]. Governments, in particular that of the United States of America, continuously repressed those

1 Wikipedia Gold Reserve Act Retrieved 2018-06-28
2 New York Times France to End Controls on Foreign Exchange 1989-12-13
3 Wikipedia Phreaking Retrieved 2018-06-15
4 Wikipedia Public-key cryptography Retrieved 2018-06-15
5 Wikipedia Cypherpunk Retrieved 2018-06-15
6 Wikipedia Hacker culture Retrieved 2018-06-15
7 Wikipedia Free software movement Retrieved 2018-06-15
8 Electronic Frontier Foundation
9 Wikipedia Electronic Frontier Foundation Retrieved 2018-06-15
10 John Perry Barlow A Declaration of the Independence of Cyberspace 1996
11 Wikipedia A Declaration of the Independence of Cyberspace Retrieved 2018-06-15
12 Wikipedia Crypto Wars Retrieved 2018-06-15

working with cryptography. The conflict, in the mid-1970's, between academics and industry during discussions around the first national standard for encryption, the Data Encryption Standard (DES)[1], established the early antagonism between the cryptography community and the government[2]. The experience of Phil Zimmermann the developer of the encryption software Pretty Good Privacy (PGP) becoming a target of a criminal investigation by the US Government in 1993 engendered the first creative response: publication of source code as a book under protection of freedom of speech guaranteed by the first amendment[3]. (This response was later repeated during the DeCSS battle with release of that source code in various printed ways and even as a song.) The experience of the first dominant browser for the World Wide Web, Netscape, over export restrictions due to inclusion of encryption code exposed the differences between government interests and user interests. The fight, between 1993 and 1996, against the attempt by the US Government to require that all encryption devices include a specific chip with a back door exposed the opposite interests of the government and the cryptographic community. The legal fights by Daniel J. Bernstein[4] in 1995 and Peter Junger[5] in 1996 over the right to publish internationally or teach foreign students cryptographic material strengthened the general distrust of the government in this community. The experience following the release, in 1999, of the DeCSS program to decrypt the encrypted content of Digital Video Disks (DVD), in which the only publicly known developer Jon Johansen was tried by Norwegian government, showed the community the problems were international[6]. This negative experience would continue even past the arrival of Bitcoin, with, for example, the exposure, by Edward Snowden in 2013, of the systematic spying by the US government on all communication, worldwide. Community experience showed that the interests of the individual could be attacked by governments.

The experience of early developers of digital currencies allowing the direct, anonymous exchange of value between end users also led to their fundamental distrust of governments. In 1983, David Chaum published the paper "Blind Signatures for Untraceable Payments"[7] proposing an electronics payment system which would neither reveal the details of every transaction, as when using a bank, nor suffer from lack of proof of payment, as when using fiat cash. This led to the formation, in 1995, of the company *DigiCash*, implementing the ideas[8]. An alternative approach for allowing interchange of value between users, using a digital currency backed by gold, was launched in 1996 under the name *E-gold*[9] which successfully grew into a widely used system. However, despite building a successful service, cooperating with law enforcement investigations, and falling outside the definition of a money transmitter, the creators of *E-gold* were prosecuted and convicted by the government, though eventually only to risible sentences. Wei Dai proposed a concept *b-money* in a 1998 publication[10] in which the electronic scarcity of money comes from solving a computationally difficult problem. In the same year Nick Szabo proposed the concept *Bit gold* where "unforgeably costly bits could be created with minimal dependence on trusted third parties, and then securely

1 Wikipedia Data Encryption Standard Retrieved 2018-06-28
2 Whitfield Diffie "The First Ten Years of Public-Key Cryptography" *Proceedings of the IEEE* 76(5) 1988-05
3 Wikipedia Pretty Good Privacy Retrieved 2018-06-28
4 Wikipedia Bernstein v. United States Retrieved 2018-06-28
5 Wikipedia Junger v. Daley Retrieved 2018-06-28
6 Wikipedia DeCSS Retrieved 2018-06-28
7 David Chaum "Blind Signatures for Untraceable Payments" in Chaum D., Rivest R.L., Sherman A.T. (eds) *Advances in Cryptology* pp199-203 Springer, Boston, MA USA 1983
8 Wikipedia DigiCash Retrieved 2018-06-15
9 Wikipedia E-gold Retrieved 2018-06-28
10 Wei Dai bmoney 1998

stored, transferred, and assayed with similar minimal trust. Bit gold."[1] Neal Stephenson, in 1999, published the widely read thriller *Cryptonomicon* whose central plot line involves developing an online mechanism for cash exchange[2]. Also in 1999, the *Gold Age* project emerged to allow the exchange of digital currency backed by gold; the company would be shut down by regulators in 2006[3]. A similar effort, *e-bullion* emerged in 2001, grew to wide popularity, and then collapsed in 2008. In 2005, Ryan Frugger founded *RipplePay* as an electronic payments system[4] in which users exchanged debt to each other. In 2006, the *Liberty Reserve* digital currency, backed by fiat currency, enabled the direct exchange of value between users[5]. The creators of *Liberty Reserve* were also prosecuted. None of these projects survived; those which were implemented all used a centralized structure and, since they were invariably partially used to launder money, were shut down by governments, generally with the originators sent to jail.

At the start of the second millennium, all of the elements were in place for the invention of blockchain systems. Individuals wanted digital cash. Individuals had computers and access to a global communication network, individuals had learned to collaborate in large distributed projects, individuals had access to a vast research literature discussing useful algorithms, and individuals trusted themselves enough, and distrusted governments sufficiently, to look for solutions on their own.

Blockchain 1.0 Era (2009-2015)

The era of blockchains started with the release of Bitcoin software in early 2009. While it would take some time for the importance of the project to be recognized, in retrospect we can see that the release of Bitcoin was of fundamental importance in that it created a community running a new type of project. In retrospect, it is useful to label as *Blockchain 1.0* the historical era from the start of Bitcoin until the emergence of newer platforms, notably Ethereum at the start of 2015, sharing a similar fundamental structure but providing functionality beyond a single digital currency.

The Blockchain 1.0 era was a time of self-discovery for the Bitcoin project in particular and for blockchain technologies in general. Bitcoin gained its first community, suffered through its first crises and coding errors, survived the first transition of leadership, built out an ecosystem of support around the project and, eventually, discovered some of its limits. This era also saw the rise of alternative blockchain projects, first as discussions and then through implementations, exploring extending blockchains from cryptocurrencies to other realms such as domain names, then to alternative currencies, and finally to more flexible systems.

Bitcoin

The Bitcoin project started from an initial release in January 2009 and grew into a massively popular project. In parallel with this growth, the project adapted to changes, both internal and external. Even as the project attempted to adapt, the inability of the project to solve all of its issues and to resolve all of the conflicting interests of its users led to an explosion of alternatives and thereby to the next era of innovation in blockchain technology.

1 Nick Szabo 1998 Bit gold Version from 2008-12-27
2 Neal Stephenson *Cryptonomicon* Avon 1999.
3 Wikipedia Gold Age Retrieved 2018-07-26
4 Wikipedia Ripple (payment protocol) Retrieved 2018-06-15
5 Wikipedia Liberty Reserve Retrieved 2018-06-28

Bitcoin arrived. The Bitcoin project was first announced[1] by its pseudonomous author Satoshi Nakamoto on a discussion forum, the cryptography mailing list[2], on the 31st of October 2008, referencing a research paper[3] hosted on the *bitcoin.org* web site[4]. Two months later, Bitcoin's first release was announced[5] linking to a source code repository[6] (although the repository has since changed[7]). Despite having been developed solely by its author, Bitcoin arrived with all the necessary components to become a community project.

Bitcoin worked. Bitcoin arrived in January 2009 as a fully formed design, with functioning code. From its initial release, Bitcoin offered a solution for digital currency: it could generate provably scarce digital resources through a mechanism requiring repeated, intensive calculations in a random search space, allow for the exchange of those resources directly between users using public key cryptographic signatures, and ensure the resources were only exchanged once through a verifiable, shared ledger of all transactions. Bitcoin, for the first time, combined data structures, computations, and network exchanges sufficient to satisfy the fundamental requirements of a purely digital currency which could be exchanged directly between peers. Even more remarkable than being fully functional at its inception, the economic incentives of the Bitcoin project were sufficiently well balanced to offer value to past, current, and future users alike, making the project economically viable as well as merely functional. Bitcoin has only suffered two major bugs in its working code[8]. The first, in October 2010, a coding error in the validation code, allowed transactions to generate currency through a numeric overflow. The first exploit of this error led to transactions generating vastly more money than was allowed by the rules of the system and so was quickly spotted. The fix required discarding the last blocks in the chain. The second, in March 2013, arose from a change in the database engine backend which led to a split in the shared database. The split was resolved by users abandoning the new code and rolling back several days worth of transactions[9].

Bitcoin grew. Bitcoin experienced exponential growth over its first decade. The number of users holding bitcoins or those participating in the network grew massively from an original small set of enthusiasts, through a larger pool of technologically savvy early adopters, to a much larger pool of general public and investment institutions. The use of Bitcoin also grew. In 2011, the dark web marketplace Silk Road emerged offering access to illegal products, mostly drugs, in exchange for Bitcoins; this greatly stimulated the use of Bitcoin. A number of foundations and businesses started accepting bitcoin as contributions or payments. The number of transactions also grew massively in that time. Similarly, the value of the cryptocurrency tokens grew from being essentially worthless to holding significant value. This spectacular growth served as its own advertising for the system, with growth reinforcing more growth.

Bitcoin adapted, both to internal changes and to external pressures.

Bitcoin's goals adapted over the years. The paper and initial code published in the name of Satoshi Nakomoto, a name widely believed to be pseudonymous, focus exclusively on the creation of a digital currency which can be exchanged directly between participants. The successful

1 Satoshi Nakamoto Bitcoin P2P e-cash paper 2008-10-31
2 Cryptography mailing list
3 Satoshi Nakamoto Bitcoin: A Peer-to-Peer Electronic Cash System 2008-10-31
4 bitcoin.org
5 Satoshi Nakamoto Bitcoin v0.1 released 2009-01-08
6 The Bitcoin project at Sourceforge
7 The Bitcoin project at Github
8 Wikipedia History of Bitcoin Retrieved 2018-07-26
9 Vitalik Buterin *Bitcoin Network Shaken by Blockchain Fork* Bitcoin Magazine 2013-03-12

demonstration of a working digital currency shifted the ambition of participants from building a digital currency to making the currency work better or to using the platform for more uses. Some users aimed to scale the Bitcoin system to support much larger rates of transactions. Other users focused on developing alternative currencies. Other users sought to apply the Bitcoin system to new uses, for instance, to enable the exchange of non-digital resources or to provide a more reliable voting system. A final group of users chose to expand the capacity of the system to integrate complex processing rules. In response to these evolving goals, Bitcoin would adopt a conservative stance leading the proponents of these new goals to develop them in alternative systems rather than within Bitcoin itself, thereby eventually leading first to alternative projects and then to the next era of blockchain systems.

Bitcoin's uses adapted over the years. While the original aim of Bitcoin was strictly as a peer-to-peer cryptocurrency, new uses emerged. The use of Bitcoin as a platform to make declaratory statements of information unrelated to the currency actually arose from the very first block. The genesis block contains a headline from a newspaper. This embedded headline served both as a political statement and as a timestamp for the genesis block, since it showed the block must have been formed *after* the publication of the newspaper headline. The ability to embed arbitrary content in the blockchain allowed for the inclusion of silly content or even vanity information. More interestingly, the ability to embed arbitrary content would be used as an internal polling mechanism for project decision making. Eventually, it was discovered that the ability to embed arbitrary content in the blockchain would allow other projects to *overlay* themselves on the Bitcoin blockchain. The other projects could record their transactions within the arbitrary content of the Bitcoin blockchain, benefiting from the processing being done to maintain the Bitcoin blockchain. A second major use of the platform, was to intermediate between the Bitcoin network and the end users. As the Bitcoin blockchain grew and the number of transactions increased, it became more and more costly for users to operate as true peers on the network. Organizations emerged to provide this processing on behalf of the users; users would trust these organizations to provide correct information about the state of the blockchain archive without needing to operate full peers on the Bitcoin network. This evolution allowed a vastly larger number of users to participate in the system, although abandoning the original goal of peer-to-peer exchange without any trusted intermediaries. The desire to develop even more uses for Bitcoin would raise the foundational question: is Bitcoin to be a platform on which these uses are to be built? The early reluctance of Satoshi Nakamoto to the proposal to build a naming system *on top of* Bitcoin, coupled with the conservative approach of key developers (and possibly commercial conflicts of interest), led to a negative answer to this question, leading Bitcoin to remain a simple system focused on the Bitcoin cryptocurrency only. This would lead, first, to the emergence of alternatives to Bitcoin, discussed next, and, then, to the emergence of new, more flexible blockchain systems in the next blockchain era.

Bitcoin's participation adapted over time. Software developers, starting with Satoshi Nakamoto, voluntarily contributed their work to the project. The lack of any financial reward for this work led developers to forming or joining companies whose business income could pay for this software development work. Participants in the processing of the blockchain transactions and the mining of new blocks moved from working independently to working collaboratively in pools, thereby trading rare, large payouts for more regular, smaller payouts. Users changed from directly participating in the network to accessing the system through third party exchanges. An entire ecosystem of

observers emerged, providing discussion fora[1,2], news sites[3,4], and web sites through which to follow the blockchain and its transactions[5,6].

Bitcoin's self-governance adapted over the years. Governance of Bitcoin originally lay with its founder, through implicit trust and because he controlled real world resources such as the `bitcoin.org` domain. In April 2011, after two years in control, Satoshi Nakamoto passed the reins of leadership to Gavin Anderson, who himself immediately attempted to distribute control among various other developers. Later that same year, the process for future development of Bitcoin software was formalized with the introduction of a formal process, the Bitcoin Improvement Process[7], modeled after the Python Enhancement Proposal process for improving the Python language. In the next year, 2012, there was an attempt to form a foundation, the Bitcoin Foundation, to govern the project but this effort failed to gain acceptance from all participants. Many of the core developers of Bitcoin eventually joined the company Blockstream[8], leading to potential conflicts of interest between the interests of the Bitcoin project and the company[9]. Currently Bitcoin operates trough a loose collaboration of its participants leading to a rather weak self-governance structure. This loose governance is held partly responsible for the regular splits in the community leading to new forks of the blockchain.

Bitcoin's technology adapted, notably both the software code and the hardware.

The code run by participants in the Bitcoin project evolved through increasingly frequent releases. The first major step was to port the code from operating only on a single operating system to operate on others, most notably the various flavours of UNIX like systems. The integer overflow bug, exploited in block 74,638 to produce an excessive number of Bitcoins, was fixed. Similarly, the bug from the switch in storage backends at version 0.8 which lead to a fork in the blockchain had to be fixed. Beside this maintenance, the code also improved to provide new functionality. The one major area of code evolution was in the rules which allowed spending an unspent pool of Bitcoins. The first method was called pay-to-public-key (`P2PK`). Since this required the public key to be published on the blockchain, it was felt that this invited efforts to attack the public keys. This led to the pay-to-public-key-hash (`P2PKH`) method where only the hash of the public key was published and the actual public key was only provided in the transaction to spend the funds. Further improvements included accepting a subset of a pool of signatures (`M-of-N`) in BIP 11, the ability for the recipient to decide the terms of spending the funds with the pay-to-script-hash (`P2SH`) in BIP 16, and even the ability to use elapsed time as part of the script requirements with BIP 65. Other changes in code included altering the internal structure of the blocks themselves with the segregated witness proposal of BIP 141. Many other proposed changes failed and participants who wished to explore such options were forced to develop them as other projects, either independently or as forks of the Bitcoin system.

The hardware used to mine Bitcoin blocks by calculating hashes also evolved in time. Originally, the hashes were calculated by the general purpose processor used as the central processing unit

1 BitcoinTalk
2 The bitcoin sub-reddit
3 Bitcoin Magazine
4 Coindesk
5 Bitcoin Block Explorer
6 Blockchain Bitcoin Block Explorer
7 The Bitcoin Improvement Proposals are hosted in their own code repository.
8 Blockstream.com
9 Rick Falkvinge Rick Reacts: How Blockstream failed and tool the BTC fork with it YouTube 2018-02-11

(CPU) on most computer systems. A first improvement was to move this work to the graphical processing units (GPU) available generally as graphic add-on cards. The next improvement in mining was to build special purpose computers for proof-of-work mining. Currently, the most productive mining systems calculate hashes using specially designed chips, termed application specific integrated circuits (ASIC). This trend towards specialization of hardware also changed the nature of participation in mining, leading from mining individually, to mining as part of a pool of participants (trading rare big payouts for regular small payouts), and, eventually, to the concentration of power in the hands of a few mining pools.

Bitcoin adapted in response to external changes, after the first two years where it was mostly ignored.

Bitcoin adapted new ways to allow outside users to obtain Bitcoin tokens. Originally, users had two ways to obtain bitcoin: through mining or through person to person exchange. The evolution in complexity of mining restricted the former while the rise in popularity of Bitcoin made impractical the latter. Bitcoin exchanges emerged in response, and the Bitcoin network moved from a homogeneous architecture to a heterogeneous system. These exchanges enabled the conversion of fiat currency to bitcoin tokens and back. While this greatly expanded the ease with which new users could obtain Bitcoin, the experience of exchanges has been problematic. Exchanges repeatedly experience theft and several have collapsed completely.

Bitcoin adapted to changes in government policy. As the weaknesses in the pseudo-anonymity of Bitcoin identities became clear, efforts were undertaken to improve the situation. The idea of tumblers emerged, an address in which many would send Bitcoin and receive the same amount in return to new addresses, thereby confusing the tracking of coins through the system. As the governments started to assert control, participants adapted. In march of 2014, the Internal Revenue Service of the United States asserted that, for tax purposes, Bitcoin would be treated as a commodity, requiring users to declare all gains made on Bitcoin as capital gains[1].

Bitcoin also discovered its limits.

Bitcoin's growth forced it to face its fundamental, structural limits. Bitcoin, as originally implemented, faced structural limits in its ability to scale the rate of transactions[2]. The combination of the ten minute average delay between blocks and the fixed maximum size of any block formed a hard limit on the transaction rate. The discussions seeking to find solutions to this structural limit, such as through the trivial step of increasing the maximum allowed size for a block, revealed a fundamental schism in the community. One group focused on building Bitcoin as a currency system focused on storing value, and therefore conservative in its changes and suspicious of any change which concentrated power with the miners. Another group focused on building Bitcoin as a currency system focused on exchanging value, and therefore was interested in making all the changes necessary to increase the transaction rate.

Bitcoin's design forced it to face its inherent inefficiency. Bitcoin, by design, wastes a vast amount of resources on the useless calculation of a 'puzzle'. This calculation translates into a massive waste of energy[3] and a net drain of money from the payment system. Suggestions to move to other modes

1 US Internal Revenue Service IRS Virtual Currency Guidance 2014-03-25
2 Wikipedia Bitcoin scalability problem Retrieved 2018-07-26
3 Wikipedia Bitcoin#Energy_consumption Retrieved 2018-07-26

of operation, such as abandoning the proof-of-work consensus mechanism, met the same schism between conservatives and progressives.

First Alternatives

As Bitcoin became established, by late 2010, the community starting exploring other uses.

The first exploration of alternative uses for Bitcoin and blockchains started in September 2010 when the community discussed the BitDNS concept to establish and exchange the ownership of names and the BitX concept blockchain to host the BitDNS and any other extension beyond Bitcoin. In the discussion of whether BitDNS should be part of the Bitcoin system directly, Satoshi Nakomoto, still active at the time, suggested that these projects proceed as separate from Bitcoin[1]. In early 2011, Namecoin, emerged as a naming system for the `bit` namespace using its own, separate blockchain.

The community also began offering alternative cryptocurrencies. The first cryptocurrencies separate from Bitcoin emerged late in 2011 essentially as copies of the Bitcoin system and using their own independent blockchain, the so called *alt-coins*[2]. Many of these projects were simple copies while others changed certain aspects of the system such as the rules through which new blocks were generated. The Litecoin blockchain project emerged using a separate blockchain in 2011. It became such a trend to release alternative coins that Dogecoin[3], which emerged explicitly as a joke, took on a life of its own, and became a viable cryptocurrency project.

The community also discussed the use of Bitcoin for the trading of other assets. Due to the particular design of Bitcoin, users realized that individual pools of Bitcoins could be assigned specific meaning outside their value as a cryptocurrency, leading to the concept of *colored coins*[4]. If some group of users decided to, they could extend the Bitcoin software to track a specific group of coins and associate a specific meaning to those coins. This design requires users to use special software aware of the colored coins. Conceptually, some system could be devised to associate these colored coins with physical objects; trading the coins would be a way to trade the associated entity. The first of these extended software wallets was released in September 2012.

Another approach to expanding the use of Bitcoin came from the development of the concept of an *overlay*[5]. This approach also requires building special software. Through experience, users had learned that it was possible to embed arbitrary data in the Bitcoin blockchain, leveraging the Bitcoin system for its ordering of transactions and its distributed storage of the ledger but serving a purpose distinct from exchange of the bitcoin currency. For some users, this was a legitimate use of Bitcoin whereas for others this was an abuse, burdening the distributed Bitcoin ledger with data that served a separate purpose. Since the existing code could not prevent the embedding of data, a compromise was reached to provide an approved way to embed data of a small size. However, with the March 2014 release of Bitcoin version 0.9.0 integrating the OP_RETURN script element, the size of the allowed embeddable data was cut from 80 bytes to 40 bytes. This change might have arisen from the commercial competition between the Blockstream and Counterparty companies[6]. The hostility

1 Satoshi Nakomoto "Re: BitDNS and Generalizing Bitcoin" 2010-12-10
2 The original meaning of *alt-coin* was any coin other than Bitcoin; this usage restricts the term to the cryptocurrency projects formed as copies of Bitcoin.
3 Dogecoin.com
4 Coloredcoins.org
5 Hal Finney "Bitcoin overlay protocols" Bitcoin Talk 2010-12-04
6 Rick Falkvinge "Rick Reacts: How Blockstream failed and took the BTC fork with it" YouTube 2018-02-11

of core Bitcoin developers to overlays would lead Ethereum to develop its project entirely separately from Bitcoin[1].

As the Blockchain 1.0 era closed, it became clear that Bitcoin would not be the single, canonical blockchain project underlying all others; the world of blockchains would necessarily involve multiple separate projects, each with their own communities, goals, structures, software, and histories.

Blockchain 2.0 Era (2015-2016)

The second era of blockchain projects arose with the arrival of second generation blockchain projects, those that supported stored procedures. This reinvention of the blockchain project, not as simply a system to support a single cryptocurrency but as a general informatic platform aiming to support multiple uses, gave rise to the second era of blockchain projects, which, retrospectively, can be called *Blockchain 2.0*. We can consider this era to start around the middle of 2015 with the release of the *Ethereum* project, the first of these second generation blockchain projects. While the technological changes brought by this era have not yet ended, we can, for convenience, consider the era to close when this project was forced into a hard fork in the middle of 2016, as a consequence of the failure of the first stored procedure with major investment, *The DAO*. The next era would emerge from the experience of *Ethereum*, from the loss of innocence due to the failure of *The DAO*, and from the innovative explosion due to discovery of the possibilities offered by second generation blockchains.

The blockchain 2.0 era was rife with promise and possibility. The continued growth in the value of cryptocurrencies and the increasing popularity of blockchain projects encouraged more investment in time and energy in these systems. The era suggested that business relations between parties could be based on a new foundation, fixed function stored procedures, which would embody the entirety of the relation and would execute its conditions without ambiguity. This led some to argue that code was the new law. Second generation blockchain projects eclipsed their first generation brethren, since the second generation could support all the functions of the first generation and do much more. Second generation blockchains also enabled users to create new, custom cryptocurrencies without needing to build, launch, and run a separate blockchain project, which led to an explosion of new cryptocurrency offerings.

Bitcoin

The Bitcoin project continued through the second era of blockchains with massive growth. The project increased in notoriety, the community gained users, and the currency rose in value.

The Bitcoin community, however, became increasingly divided over how to improve the project. The question of whether Bitcoin would try evolve into a much more flexible system had been answered in the negative, when Ethereum had been created as a separate project. However, the question of how to improve the performance of Bitcoin itself was unresolved.

The first forks of the Bitcoin project emerged as different groups sought to improve the performance by increasing the overall size of each generated block. The *Bitcoin XT* project emerged in August 2015, the *Bitcoin Unlimited* project in December 2015, and the *Bitcoin Classic* project in

1 *ibid*

February 2016. While none of these projects would survive in the longer term, they did show that forks of the Bitcoin project were possible and did not even much affect the value of the Bitcoin currency, thus opening the door to future forks.

Ethereum

The Ethereum project[1], intending to build a blockchain system with fully flexible transaction scripts, was proposed in 2013, formed and funded in 2014, and released in 2015[2]. Whereas in Bitcoin the scripting language only allows a limited set of operations and the blockchain validation code only accepts a small set of allowable types of scripts, Ethereum wanted to extend scripting to allow any operation. The Ethereum scripting language was to be Turing-complete, that is, logically as flexible as the major programming languages. As the design developed and worked progressed, the Ethereum project realized it would require three major components: the blockchain system with its *soliditiy* scripting language, a messaging protocol between network nodes, the *whisper* protocol, and a storage protocol, *swarm*.

The release of the Ethereum project opened the door to a massive new wave of innovation *on top of* a blockchain project rather than *as a new* blockchain project. The ability to create scripts of arbitrary complexity meant that anyone could design new functionality. These scripts which run on the blockchain are sometimes referred to as *chaincode*, in order to differentiate them from the regular code of the blockchain itself.

Chaincode permitted the creation of new cryptocurrencies which run on the blockchain rather than being part of the chain. Where Ethereum coins (and the related gas coins) are integral to the blockchain code itself, the new type of cryptocurrency could be built as standalone code to be run by the blockchain system. The units of these cryptocurrencies build on top of the blockchain are often refered to as *tokens*[3], as distinct from *coins* which are units of cryptocurrencies integral to the blockchain, although this terminology is not followed consistently. This approach to cryptocurrencies led to an explosion of new tokens and, eventually, to a standard, called ERC-20[4,5], to streamline these cryptocurrency systems and facilitate the exchange of their tokens. There are currently over a hundred thousand such tokens on the Ethereum blockchain[6].

Chaincode also permitted the creation of more complex systems, including systems able to acquire and hold funds then to decide how to use the funds, and to spend the funds. The first major *decentralized autonomous organization* of this type, called *The DAO*, would capture almost one sixth of all Ethereum coins in circulation, a value of around $150 million at the time[7]. The enthusiasm of this effort led to arguments that these new chaincode based scripts would usher a new type of contractual relationship between parties, determined only by the computer code in the script, leading to the increased use of the name *smart contract* for these scripts and to the conceptualization of 'code as law'. However, the DAO failed spectacularly: as the month long funding campaign was winding down, a vulnerability was found in the script, and this vulnerability was used to drain around a third of the stored funds. The collapse of the DAO would have fundamental consequences. First, the collapse split the Ethereum community into those who wanted

1 Ethereum.org
2 Wikipedia Ethereum Retrieved 2018-07-25
3 Ethereum Create your own crypto-currency with Ethereum
4 ERC20 Token Standard
5 Wikipedia ERC20 Retrieved 2018-07-25
6 Etherscan Token Tracker
7 Wikipedia The DAO (organization) Retrieved 2018-07-25

to stick to the original concept of letting the code of the script be the final arbiter of what was allowed and those who wanted to refund all the funds committed to the DAO to the original contributors. This split would lead to a fork of the Ethereum project into *Ethereum Classic* which accepted the original ledger despite the hack and the new *Ethereum* which reverted all the transactions involved in the failed DAO. Second, the collapse led to a more mature discussion of code based contracts; it was apparent forever after that chaincode scripts, or smart contracts, might require external supervision. Third, the collapse would lead to the end of the second era of blockchains as it became clear that a much more rigorous approach would be required to ensure these systems did not collapse due to coding errors.

Experience

The community continued to be affected by external forces. For example, in august of 2015, the Department of Financial Services of the State of New York issued the first rules regulating exchanges, requiring that businesses obtain a BitLicense in order to have residents of that state as customers[1]. This has led some businesses to adapt their operations to obtain that registration. Other jurisdictions were slowly establishing new rules for cryptocurrencies.

As the second era of blockchains came to a close, it became clear that the world of blockchains would be even more diverse and complex that previously imagined. Not only did Ethereum not replace Bitcoin as the dominant canonical blockchain but Ethereum was not even to remain a single project. The Bitcoin project had split into multiple forks as had Ethereum. The stage was set for major innovation in multiple directions: through new blockchain projects, through forks of existing blockchain projects, and through new uses of chaincode in existing second generation blockchain projects. Innovation would also come from the development of generic code for blockchain projects and through the use of that code in specific, restricted communities.

Blockchain 3.0 Era (2016-2018)

The third, and current, era of blockchain projects can be called, for convenience, the *Blockchain 3.0* era and can be considered to start with the hard fork of the Ethereum project in the middle of 2016 which intended to repay the investors in The DAO. Unlike the two earlier eras, this era does not arise with the release of a technologically innovative project. Instead, the era encompasses multiple efforts in many new directions, all infused with a realistic view of the world of blockchains.

The blockchain 3.0 era is one of creative realism. While the blockchain projects developed in earlier eras continue to run and the cryptocurrencies they support continue to work, they are increasingly forced to face reality from investors, judiciaries, regulators, and users. New blockchain projects being built, rather than attempting to build more complex platforms, are attempting to replicate second generation blockchain systems while avoiding their problems. Collaborative projects have emerged to enable the generation of custom blockchain projects for specific business goals. The support for arbitrary scripting by second generation blockchain projects has provided a new kind of computational platform, leading to an explosion of creativity. Projects have arisen to build an entirely new type of software, the so-called *distributed applications* or *Dapps*, which use blockchain projects to provide their backend services rather than relying on services built with a traditional design. This current explosion of innovation is one reason the world of blockchains is of

1 New York State, Department of Financial Services, BitLicense Regulatory Framework, 2015.

such current interest; at the same time, the explosion of complexity makes it impossible to fully encompass current innovations.

Ongoing blockchain projects

The most important existing blockchain projects continue to operate. However, these older projects face increasing competition from other projects and even from forks of themselves.

The original Bitcoin project is perhaps reaching its limits.

The exponential growth in value may have finally stalled in mid-2017 reaching a peak of almost $20,000 before falling back to around eight thousand dollars today. This has been coupled with an increased examination of how the Bitcoin market establishes the value of the token leading to widespread suspicion of market manipulation[1], including judicial review[2]. Because of the massive expenditures on hardware and electricity necessary to maintain the Bitcoin system, as a whole the Bitcoin economy looses value constantly, raising questions on its ultimate ability to maintain itself.

The exponential growth in transaction rate eventually ended, from a high of around 400 thousand per day in mid 2017 to around half of that currently[3]. This is partly due to rise in fees and partly due to scaling issues with the Bitcoin design. As the value of Bitcoin rose, the transaction fees rose in lockstep. This led to a fall in the number of transactions and changed the nature of transactions since transactions for small amounts ceased to be cost effective. More critically, the Bitcoin design, which combines an average interval between blocks of around ten minutes with a maximum block size restricting each block to around 5000 transactions, means that the system is inherently restricted to around ten transactions per second. This means that any competition between users trying to get transactions accepted will lead directly to higher fees. The costs of transactions have become a significant barrier to further growth, leading to many proposals for change, most centered around raising the limit on the size of each block.

The failure to adapt the Bitcoin system to the rise in transaction fees and to the issues of scaling has forced participants away from the original code base. Newer, more recent forks continue to split the Bitcoin community. In 2017, both Bitcoin Cash (BCH) and Bitcoin Gold (BTG) forked from Bitcoin and many future forks are planned.

Similarly, the Ethereum forks, Ethereum Classic and Ethereum (new), both continued to function.

New blockchain projects

New blockchain projects continue to emerge. Some merely follow earlier design paths and are only of real interest to their participants. However, a third generation of blockchain projects is also emerging, focused on addressing the limitations of current projects.

The *Cardano* blockchain project[4] is one of the newer blockchain projects. Cardano aims to build a second generation blockchain project, that is one that supports arbitrary scripting, just like Ethereum. However, rather than being focused uniquely on the functionality of the system, Cardano developers focus also on the development process and on solving issues encountered by previous

1 Neil Gandal *et al.* Price manipulation in the Bitcoin ecosystem *Journal of Monetary Economics* Vol. 95, pp,86-96, 2018-05
2 Matt Robinson and Tom Schoenberg U.S. Launches Criminal Probe into Bitcoin Price Manipulation Bloomberg 2018-05-24.
3 Blockchain.com Confirmed Transactions per Day
4 Cardano.org

projects, such as code quality and governance. For example, the Cardano blockchain system is being written using the Haskell programming language which is not as popular as more mainstream languages but is able to be verified formally through code verification tools. Similarly, Cardano development is accompanied by academic publication to facilitate the review of proposed ideas and the work is accompanied by extensive audits by third parties. This slow and deliberative approach is designed to ensure that the Cardano system, once it is built, is more robust than earlier projects. The presentation of Cardano by Charles Hoskinson[5], one of the project's founders, explains both his take on the history of blockchains and the development philosophy of the project developed under peer review and focused on high assurance code.

Blockchain generator projects

The current era includes other types of blockchain efforts which aim, not to build a single network with one shared blockchain, but to develop software code with which users can build their own, custom blockchain projects. The vision of these efforts is that, since certain communities will want to run a blockchain project for their own aims, those communities will need robust, customizable software capable of generating blockchain processing systems.

The Linux Foundation has started the Hyperledger project[3] as an umbrella which can host the tools necessary for groups to build blockchain generating software. Since these efforts are still experimental, the Hyperledger project is accepting a number of different projects each with different approaches. For example, Hyperledger includes multiple blockchain generators, Fabric, Indy, Burrow, Sawtooth, Iroha, each with different characteristics, but Hyperledger also includes a number of projects to develop tools such as Composer which provides a simple domain specific language to generate the information system logic of the project without needing to understand the details of each implementatation.

Chaincode projects

The current era of blockchains includes a large amount of work on code to be embedded as stored procedures in the newer blockchains, the so called *smart contracts*.

The recent boom in initial coin offerings (ICOs) relied heavily on stored procedures for tokens, mostly on the Ethereum blockchain. These ERC-20 compliant tokens could be held and traded just like other cryptocurrencies. Many new projects formed and funded their work by selling or auctioning new tokens.

One recent experience with these stored procedures was the recent CryptoKitties craze. These are collectible non-fungible (*i.e.* unique) ERC-721 compliant tokens which generate stylized pictures of cats and which can interact to generate new unique token instances. The popularity of these tokens, at their height, accounted for a major part of traffic on the Ethereum blockchain system.

One major other focus of procedures stored as chaincode has been to act as a backend for distributed applications, or Dapps. The idea is that an application, on the desktop or on mobile phones, could use a blockchain procedure to provide services or coordination. The current reality seems to require both a blockchain backend and a regular server backend, for performance reasons.

5 IOHK IOHK | Cardano whiteboard; overview with Charles Hoskinson 2017-10-26
3 Hyperledger.org

The excitement of this approach is that it abstracts away the blockchain project itself but benefits from the replication, non-censorship, and permanence which the blockchain project provides.

Experience

General society, outside the world of blockchains, has had a growing impact on blockchain projects mostly due to the rise in popularity and value of the cryptocurrencies. Initially, outsiders were mainly a source of new participants, with the rest of the world paying no attention. As demand grew, exchanges emerged to enable participants to obtain cryptocurrencies with ever increasing ease. The traditional financial sector also responded as participants, as competitors, as investors, and as developers of new projects. Governments also eventually responded with interpretations, regulations, and legal interventions. For example, by September 2017, China would ban the raising of funds through initial coin offerings (ICO), further restricting the world of blockchains.

Within the blockchain communities, there is a growing realization that some proactive effort of working with the governments of nation states can prevent the overburdensome regulations which come when governments attempt to control an incompletely understood technology.

The future of blockchain projects currently seems promising in spite of the issues with cryptocurrencies. A more realistic mindset is taking hold where blockchain projects are not the solution of the moment to every problem. Instead, a vibrant, self-aware community is tackling many of the core issues with vigour and developing a whole new class of applications.

Blockchain, the Data Structure

In our review of the world of blockchains, we can focus on the actual blockchain, that is the chain of blocks of binary data, often used as a shared ledger in a blockchain system. However, before discussing the data structure itself, it is necessary to present cryptographic algorithms which are used by the data structure.

Cryptographic Primitives

Blockchains and blockchain projects depend on fundamental cryptographic primitives including cryptographic hash functions to generate hash values and a cryptographic signature system to generate signatures using secret keys which can be validated with public keys. Surprisingly, the simpler blockchain projects do no rely on any other cryptographic primitives, notably lacking any use of encryption. Blockchains rely on cryptography to guarantee the integrity of the stored data and to demonstrate ownership of resources.

Hash functions

Hash functions are algorithms which generate, for an arbitrary stream of input bytes, an output of a fixed size. To be useful, the output of a hash function should be different only when the input is different. *Cryptographic hash functions* are hash functions with specific properties which provide mathematical guarantees related to the function, for example that the chance of two different inputs producing the same output be essentially null.

Cryptographic hash functions play a central role in blockchain projects. Cryptographic hash values chain together the blocks of data in the blockchain, with the cryptographic properties of the hashes forming the basis for shared trust in the overall database. Hashes also perform several other uses: they help structure the contents of each block into an efficiently verifiable structure, hashes serve to testify to the existence of public keys without exposing the key itself, and hashes serve in the proof-of-work consensus mechanism used by many blockchain systems. The cryptographic hash functions used in blockchain projects have a series of properties, both practical and mathematical, useful to work at scale and necessary for establishing trust.

A generic hash function is a computer based algorithm which processes blocks of binary data of variable size and generates an output value of fixed size. Depending on the properties of the generated number, such outputs can be used for many purposes such as for checksums which help guard against data corruption. A cryptographic hash function serves as a hash function with stricter properties. Cryptographic hash functions also must be efficient enough, in memory use or computational steps, to work on the hardware for which they are targeted. Cryptographic hash functions must satisfy a number of mathematical properties:

- **limited** - the function must produce hash values of limited size,
- **deterministic** - the function must produce the same hash given the same input,
- **non-invertable** - given an output value from a hash function, $H = hash(m)$, it must not be possible to obtain the original input, m,
- **uniform** - the function must produce hash values over the entire output range,
- **non-contiguous** - the function must produce very different outputs for even trivially different inputs (also known as the *avalanche effect*),
- **pre-image resistant** - given a hash value H, it must not be possible to find any m such that $hash(m) = H$,

- **second pre-image resistant** - given an input *m*, it must not be possible to find any other input *n*, such that *hash*(*m*) = *hash*(*n*), and
- **strong collision resistan**t - it must not be possible to find any pair of input values (*m*,*n*), such that *hash*(*m*) = *hash*(*n*).

Also, a cryptographic hash function should have a number of other properties, such as resistance to length-extension attacks; the complete set of requirements for a good cryptographic hash function require extensive presentation. Note that most of these properties are transitive, applying equally to *hash*(*m*) and to *hash*(*hash*(*m*)); the properties apply to a hash value calculated from an element, or to a hash value calculated from the hash value of an element, without loosing any of the guarantees offered by the cryptographic hash algorithm. Such indirection provides a useful way to gain efficiency and so is used frequently in blockchain systems.

While there are a large number of hash functions, only a handful have achieved general use. Some of these, such as MD5, SHA-1, and SHA-2, were popular but have recently been deprecated because weaknesses have been found. The current set of popular hash functions include the RIPEMD family, SHA-256, SHA-3, and BLAKE2. The Bitcoin project uses both the SHA-256 and RIPEMD-160 hash algorithms, while the Ethereum project uses Keccak-256 and Keccak-512 algorithms (similar to SHA3_256 and SHA3_512).

Hash values are used extensively in the blockchain data structure. The blocks of the chain are linked through hash values while the contents of each block are structured into trees of hash values. The hash values provide archival guarantees that the data has not been altered, either through error or through intent. Hash values serve to obfuscate the actual public keys assigned as owners of assets. Hash functions are also used extensively in blockchain computational projects. In blockchains based on a proof-of-work block generation systems, hash functions act as the puzzle mechanism which allocates the right to generate a block to anyone solving a computational difficult task.

Public key cryptography

Public key cryptography is the revolutionary, asymetric approach to cryptography which was discovered and first publicized through the work of Whitfield Diffie and Martin Hellman, Ralph Merkle, Ron Rivest, Adi Shamir, and Leanard Adleman during the 1970s[1].

Public key cryptography enables trustworthy digital signatures and message encryption. Both of these require the generation of a pair of keys, one to be kept secret, the other to be shared publicly. Digital signatures can be produced for a given message using the secret key and the generated signature can then be validated using the message, the signature, and the public key, thus attesting that the signature was produced by the holder of the secret key matching the given public key. Message encryption can be produced using the public key to produce an encrypted message and the encrypted message can be decrypted using the secret key. Note that both of these processes are unidirectional. The holders of the secret key can sign messages and decode encrypted messages while the holders of the public keys can validate signatures and encode messages. In order for these processes to work in the other direction, a second set of key pairs must be generated and the public key of those pairs shared.

Public key cryptography plays a central role in in blockchain projects. Public keys serve as identifiers of accounts. Digital signatures are added to transaction requests in blockchain payment systems to attest both that the signer has control of the secret key for the account, thereby

1 Wikipedia Public-key_cryptography Retrieved 2018-07-26

establishing ownership, and that the account owner is the one making the request to transfer the resource. Surprisingly, encryption is not actually used in the original Bitcoin project. However, encryption can be used in more recent blockchain projects.

Key pairs

Public key cryptography starts with the generation of a pair of keys. The *key generation algorithm*, *G*(), produces a pair of keys

$$G() \Longrightarrow \{\text{publicKey}, \text{secretKey}\}$$

one to be kept secret, the other to be shared publicly.

The method for sharing the public key depends on each project. Originally in the Bitcoin blockchain, the public key served as the identifier of the account holding funds and was published onto the blockchain data archive when funds were transferred to the account. This was later changed so that only a hash of the public key was used as the identifier of the account and published on the blockchain to act as a proof-of-existence for the public key; the public key would only be revealed with the transaction to use the funds in the account. In permissioned blockchains which use public key cryptography for access control, public keys are often stored in a certificate authority and may have complex life cycles including mechanisms for key revocation.

Signature functions

Public key cryptography enables the signing of messages using *cryptographic digital signature algorithms*. These digital signatures establish that the generator of a signature which validates against a given public key also has control of the secret key matching the public key in the key pair.

Blockchain projects rely on *cryptographic digital signature algorithms* using the *key pairs* from *public key encryption* schemes. Public keys, or cryptographic hashes of these keys, can serve as the identifier of each account. If the hashes are used, the actual public key will be revealed as part of the transaction request intending to spend the resources in that account. Therefore, it is generally recommended not to reuse accounts. A signature provided with the transaction request simultaneously attests that the signer has access to the secret key and therefore owns the account and attests that the owner wishes to use the resources in the account.

Digital signature schemes, as used by blockchain projects, require two interrelated algorithms, one for signature generation $S(\cdot)$, and one for signature verification $V(\cdot)$, both relying on the keys generated by the *key generation algorithm*.

The *signature generation algorithm* requires as input both a message and the secret key and results in a signature.

$$S(\text{message}, \text{secretKey}) \Longrightarrow \text{signature}$$

The message is generally considered as a stream of bytes without any inherent meaning and most signature generation algorithms can handle messages of arbitrary size. The signature is a stream of bytes of fixed size.

The *signature validation algorithm* requires 3 inputs, the message which was signed, the signature, and the public key, and produces a Boolean as output either validating or not validating the signature.

$$V(\text{message}, \text{signature}, \text{publicKey}) \Longrightarrow \text{boolean}$$

A validated signature indicates that the signature was generated by the secret key complement to the public key used as the input.

The cryptographic properties of a digital signature provide certain guarantees. Since the public key, message, and signature are shared, the cryptographic properties of the system guarantee it is difficult (*i.e.* impossible) to derive the private key from these three elements. As long as the secret key is kept secret and under the control of the generator of the key pair, a message accompanied by a public key and a signature guarantees authentication, integrity, and non-repudiation. The authentication guarantee ensures that the message comes from the generator of the public key. The integrity guarantee ensures that the message remains as it was originally sent by the generator of the public key. The non-repudation guarantee ensures that the generator of the public key cannot claim the message was sent by another party. Note that since these guarantees rely on total control of the private, secret key by its creator, systems relying on digital signatures generally rely on a way to manage keys, for instance enabling key revocation to invalidate the key when control of the key is lost.

Blockchain projects, since the release of Bitcoin, have relied on elliptic curve cryptography (ECDSA) for their digital signature algorithms. The digital signature algorithm for Bitcoin uses ECDSA based on the Secp256k1 standard[1], section 2.4.1, which recommends the Koblitz elliptic curve $T = (p, a, b, G, n, h)$,

$$y^2 = x^3 + 7$$

where $a = 0$ and $b = 7$ along with prime Modulo,

$$p = 2^256 - 2^32 - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$$

base point (in compressed form),

$G =$ `02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798`

order *n* of *G*

$n =$ `FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141`

and cofactor

$$h = 1$$

The Ethereum project also uses this same curve and parameters for its digital signatures.

Encryption functions

Public key cryptography enables the encryption of messages using *cryptographic encryption algorithms* which rely on the *key pairs* from *public key encryption* schemes. Encrypted messages can be generated by anyone using a public key but can only be decrypted by the holders of the secret key matching the public key in the key pair.

Some blockchain projects rely on *cryptographic encryption algorithms* to store information in the blockchain which can be demonstrated to have occurred at the time it was recorded but whose content is only known to certain parties. The *Hyperledger Fabric* blockchain system uses encryption to keep certain parts of transactions secret while publicly acknowledging the transaction itself.

1 Brown, DRL Standards for Efficient Ctryptography *SEC2: Recommended Elliptic Curve Domain Parameters* version 2.0 Certicom Research 2010-01-27

Message encryption schemes, require two interrelated algorithms, one for message encryption *E*(), and one for message decryption *D*(), both relying on the keys generated by the *key generation algorithm*.

The *encryption algorithm* requires as input both a message and a public key and results in an encrypted message.

$$E(\text{message}, \text{publicKey}) \rightarrow \text{encrypted\_message}$$

The message is generally considered as a stream of bytes without any inherent meaning and most encryption algorithms can handle messages of arbitrary size. The encrypted message is another arbitrary stream of bytes.

The *decryption algorithm* requires as input both the encrypted message and the secret key and results in the original message.

$$D(\text{encrypted\_message}, \text{secretKey}) \rightarrow \text{message}$$

Note that these public key encryption schemes are unidirectional: anyone with access to the public key can encrypt messages which only the holder of the secret key can decrypt.

Other functions

Current research is exploring the use of other cryptographic functions in blockchain projects.

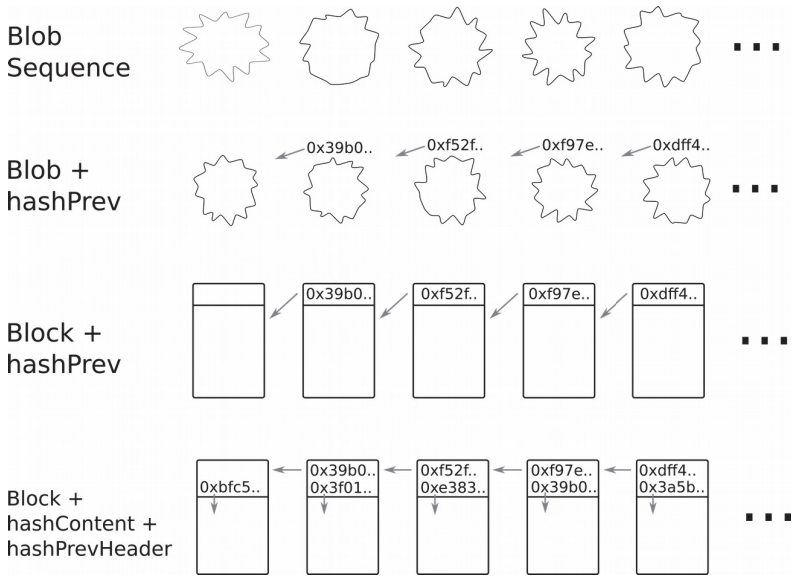
Many projects now allow for m-of-n signature systems which assign ownership of assets collectively to multiple public keys and which require signatures valid against some subset of those keys to validate and allow a transaction.

Similarly, current research on the use of zero-knowledge proofs could potentially allow users to demonstrate that something has happened, for example that a transfer of funds has taken place, without revealing exactly what happened, such as the value of the funds transferred. This is an active area of research.

The Blockchain Data Structure

The *blockchain* data structure lies at the heart of blockchain projects because it is used as the shared record of system events. A blockchain serves as the shared database and shared ledger for cryptocurrency systems built on a blockchain.

A blockchain is a remarkably simple data structure, similar to a linked list.

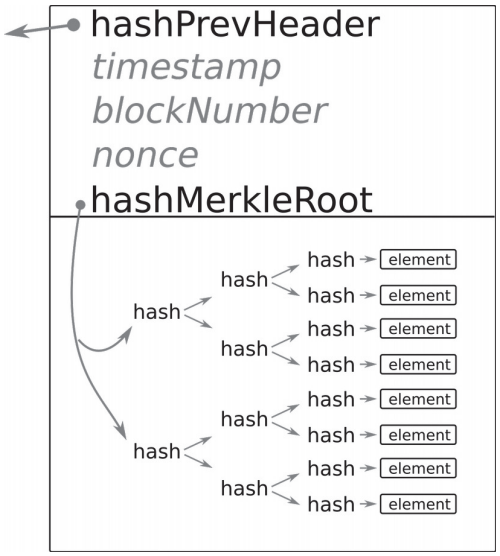


At its most fundamental, the blockchain data structure consists of an ordered sequence of chunks of data, a sequence of binary blobs.

To provide cryptographic security, the sequence can consist of tuples, each with one blob and a cryptographic hash of the previous blob. By formally integrating the hash with the blob we obtain a structured *block* consisting of a header and the content, where the header holds the hash of the previous block.

As a next step, we can integrate, in the header, a hash of the content. Due to the transitive nature of cryptographic hashes, the inclusion in the header of a hash attesting to the block content means that the chain can become a chain of headers with the hash of the previous header indirectly testifying to the integrity of the previous block's content.

Thus we have gone from a sequence of blobs of data, to a cryptographically secured sequence of block headers, where each header includes a hash testifying to the integrity of the block's content.



The blocks of actual blockchain data structures include, in the block header, some optional, but useful, information such as the timestamp when the block was created, the sequence number of the block in the chain, and possibly a nonce, a random number chosen to fulfill certain properties and generated as part of the block creation process.

A further modification brings us to the general structure of blocks, as currently used in blockchains. The hash of the contents, rather than being calculated on the entire binary stream of the content, is calculated from the two top nodes of a binary tree of hashes, a Merkle tree, whose leaves are the elements which make up the content of the block.

This structure, of the blockchain, of the blocks in the chain, and of the header and contents of each block, facilitates the use of the blockchain. The modular structure minimizes the binary size of the elements involved in the calculation of each hash value at the minor cost of calculating a few more hash values. The modular structure also allows computers which are not acting as archival storage systems to discard much of the content while retaining the hash values to be able to validate the integrity of the rest of the blockchain. The extra costs of this complicated structure is fully compensated by the flexibility the structure provides.

Actual blockchain projects can use much more complex structuring of the blocks. Some projects, like Ethereum, include multiple Merkle trees with different information. Similarly, the Merkle tree might be stored in compact form in order to save space. Some projects, such as NXT and IOTA, skip the chain of blocks and store the archival content directly as nodes in a directed acyclic graph[1].

The integrity guarantees of the hashes in a blockchain data structure make the structure suitable for archival storage. Given the hash values in the header of the last block of any blockchain, the content of the entire chain can be validated. The cryptographic guarantees of the hash functions mean that it is mathematically infeasible to alter the contents of the chain and still have the chain validate against the final value. A blockchain is therefore a useful data structure to be used as a distributed database: different users can know that they have the same database as long as the values of the hash of their last header is the same and the chain as a whole validates.

1 Sherman Lee Explaining Directed Acyclic Graph (DAG), The Real Blockchain 3.0 *Forbes* 2018-01-22

The blockchain data structure enables the storage of arbitrary binary content whose archival integrity is guaranteed by the value of the hash of the last header. These data structures can be modified only by appending new data blocks to the end of the chain. As such, blockchain data structures can grow but can not shrink. The unbounded growth of blockchains impacts the space necessary to store the archive, the bandwidth necessary to exchange the archive, and the processing power and memory required to validate the archive. One area of current research involves methods of cryptographically secure *pruning* the blockchain data structure to generate a new, smaller data structure which retains the archival guarantees of the original blockchain.

Distributed computing projects can use blockchain data structures as the shared database of past history. Each such computing project will have to define what information is stored in the blockchain data structure. Traditionally, the computing projects only stored transactions establishing a new ownership for resources. More recently, second generation computing projects enable the storage of executable computer code. The rules to validate such transactions will be specific to each computing project. Distributed computing projects have to decide on how the blockchain data structure can be extended with new information: who gets to decide what information is included, who generates a new block, and how everyone agrees that the new block becomes part of the shared history. This is generally called the *consensus* mechanism.

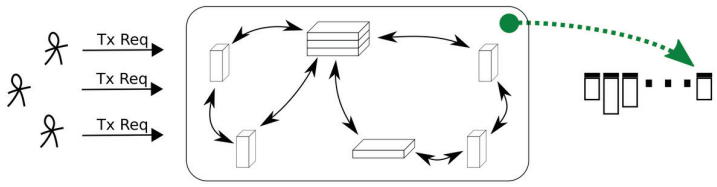
Blockchain, the Computation

The world of blockchains also includes the concept of blockchains as collaborative, distributed, sustained computations. These computations exchange and process data using computers, networks, and computer code built and run by the people involved in the community. In this view, the blockchain project is an ongoing computation in which users submit transaction requests and the computation periodically extends the shared blockchain data structure to include new transactions. As a consequence, the blockchain project must develop, maintain, and possibly extend the code running the computation. More critically, the project, despite being distributed by design, must establish a common form of governance to agree on how to build and run the computational system.

For our purposes, a working definition for a blockchain *project* could be:

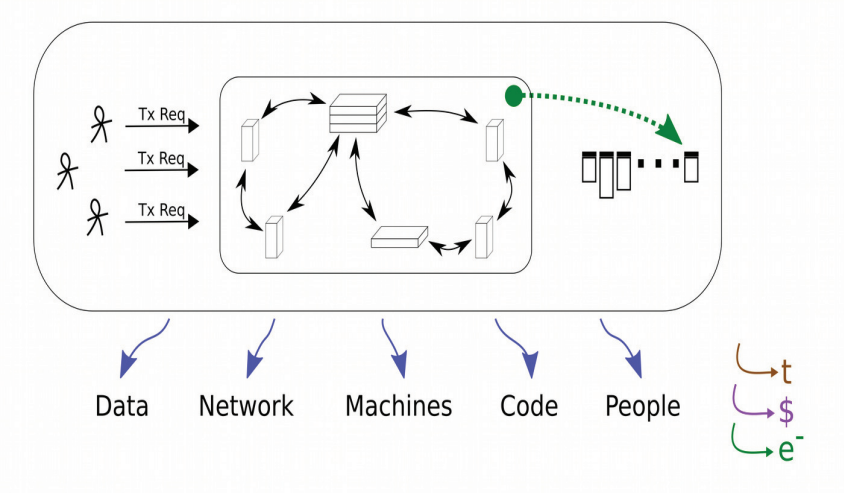
> A *blockchain project* consists of a collaborative effort maintaining an ongoing *computation* in which a *community* of participants maintains a *network* of *computers* that run *code* to exchange and process a series of *data structures* including at least one extensible, cryptographically secured, data structure which serves as the basis of shared trust.

A blockchain project, when viewed as an ongoing computation, is an information system which accepts incoming transaction requests, distributes the transaction requests across the network, and, through some mechanism, periodically generates a new block which includes the results of some of the transactions.



Since this new block might have created an incidental fork of the shared blockchain data structure, the system also includes some mechanism through which all participants eventually settle on one particular chain of blocks as the shared archival record.

A blockchain project viewed as an ongoing computation can be examined from a multitude of perspectives: as a human activity, as an economic system, as a legal jurisdiction, as an ecological system. Here, we focus on the component elements of blockchain projects. In the next chapter, we will examine the information flows of these systems.



A blockchain project, structurally, consists of data flowing through the computation, of a network enabling the exchange of messages, of machines to process and store information, of software code to perform those operations, and of the people involved in all phases of the project. Maintaining all of these requires time, money, and electricity.

The data elements of a blockchain project include the global information flow, from transactions to database, and the exchange of messages within in the network. The overall semantic meaning of the global information flow is particular to each project, as discussed in the next chapter.

The network consists of the connections between the nodes (machines) performing the calculation. The topology of the network, the bandwidth of its connections, and the pattern of messaging determine fundamental properties of the network such as the propagation time for messages. Blockchain project networks can be comprised of identical nodes throughout or be composed of many different kinds of nodes, each specialized for different activities.

The machines acting as network nodes contribute processing power, memory, and storage for the ongoing computation.

The software code involved in the computation includes code for relaying messages on the network, code for validating transactions and blocks, code to use the network, code to process network messages and generate new blocks for the blockchain database, and code to inspect the database and monitor the network.

The people in the community running the blockchain project act in different capacities, as users of the computation, as operators of machines, as developers of the code, as coordinators of the community, and even as outsiders: potential participants and observers.

Each of these is discussed, in turn, below.

Data

A blockchain project processes data structures from user submitted transactions to a project defined database. The distributed project requires the exchange of messages over the network to sustain the collaborative computation. The transactions submitted by users provide the core functionality of the system. The database, shared between all participants, forms the foundation of the project.

Messages

Blockchain projects require the exchange of data over the network through *messages*. These messages include the transaction requests being submitted to the system and the new blocks generated by the system. However, messages also are required for bootstrapping nodes or updating nodes which have been temporarily disconnected. In both these cases, nodes will need to exchange requests for past blocks and the blocks themselves.

The structure and content of the messages depend entirely on the design and operation of the specific blockchain project. The binary form of the messages will also depend on the specific blockchain project, ranging from self defined binary blobs, to HTTP like text blobs with headers and content, to remote procedure calls, such as JSON-RPC[1].

Transactions

Blockchain projects build their shared database from a series of element data structures. *Transactions* modify the shared record of system events, that is the blockchain. Generally, transactions are what gets recorded as elements of the blockchain. The term comes from database systems and refers to an atomic change of state in which either the change of state fails or the change of state is applied completely.

However, the term *transaction* is used loosely and can refer to any of several related data elements: the request for a transaction to occur (*tx_request*), the calculated result of the request (*tx_result*), and the record stored in the shared archive (*tx_record*). The loose language arises because in the first blockchain system, Bitcoin, these were all equivalent: valid requests would be included in a new block as they were, so the request, result, and record were identical. However, more recent blockchain systems allow the result of a transaction request to be non-deterministic. For example, a transaction request might be a bid that, while accepted, ends up only partially fulfilled.

The types of acceptable transactions differ depending on what the particular blockchain system allows, ranging from simple declarations to automated code.

One type of transaction simply allows for recording a *declaration*. A major use for such declaration are recording an arbitrary hash value from some other system to serve as a proof-of-existence of some digital resource at the time the block gets incorporated into the blockchain. Declarations also allow interaction between blockchains, such as the anchoring or overlay designs discussed elsewhere. Note that one critique of public (permissionless) blockchains comes from the danger of allowing declarations of arbitrary data to be recorded on the blockchain; this opens an avenue for sabotage of the system by recording in the shared record content which is repugnant or even illegal[2].

A second set of transaction types relate to assets: transactions declaring *new asset types*, transactions declaring *asset creation*, transactions declaring *asset transfer*, and transactions declaring *asset destruction*. Bitcoin only formally allows two of these, a 'coinbase' transaction to create assets as part of the mining process and a regular transaction to transfer funds. (Asset destruction only could be performed through a hack by transferring control to an address which was

1 Wikipedia JSON-RPC Retrieved 2018-07-30
2 Matzutt, R *et al. A Quantitative Analysis of the Impact of Arbitrary Blockchain Content on Bitcoin* Financial Cryptography and Data Security 2018.

provably unable to spend the funds). Other blockchain projects allowed for the declaration of various asset types and their transfer, and possibly even their destruction.

Another new type of transaction arose from the Ethereum project, the smart or *automated contract*. These are transactions which include programmatic code which reacts automatically when receiving messages from other transactions. For example, the Ethereum project enables the declaration of new asset types through the creation of specific contracts. Ethereum even formalized, in the Ethereum request for change ERC-20[1,2], one way to declare new assets such that they could easily be included in asset exchange platforms.

More complex transaction types can also exist, for instance the Hyperledger Fabric project allows transactions which change the policy rules which govern the processing of transactions.

Blockchain Database

A blockchain project uses some data structure, usually a blockchain, as a shared, replicated database. This database defines the state of the system, either explicitly or implicitly as the computational result of processing the database according to project rules. The cryptographic guarantees of a blockchain data structure ensure that, given a latter block in the chain, anyone can validate the chain up to that block. Cryptography gives archival guarantees for the data.

The blockchain project, however, must establish how the shared database is extended, that is how new blocks are appended to the blockchain. The project must have rules which establish which contents are to be included in a new block, how a new block is generated and who generates it, how this block is distributed through the network, and, in projects in which blocks generation can happen concurrently, how to choose between alternative blocks. This process of extending the collective archive is called the *consensus* mechanism.

The block generation mechanism used by Bitcoin and other major blockchain projects is called *proof-of-work* which involves all participants in searching for the solution to a puzzle and the first to find a solution wins the right to generate a block. This mechanism involves a huge amount of redundant, wasted calculations leading to a massive financial cost and environmental impact[3,4]. This inefficiency has led to the search for other consensus mechanisms which would not be as wasteful. One approach would be to randomly select a node and give it the right to generate the next block. However, since many projects place no restrictions on participation, allowing open participation in such a lottery exposes it to a *sybil attack* in which a user generates a vast number of identities which overwhelm the system. Instead, the *proof-of-stake* mechanism has been developed which gives every participant a chance to generate a new block based on the amount of an asset held by that participant. The Cardano project, for example, is establishing the Ouroboros protocol[5] as its proof-of-stake mechanism. While this approach does give larger holders greater probability of generating the next block, the mechanism can operate solely from the contents of the shared database archive. Another approach, useful for less popular projects, is of merged mining where a single proof-of-work calculation can satisfy several cryptocurrencies at once.

1 ERC-20 Token Standard
2 ERC20 Token Standard
3 Christopher Malmo Bitcoin is Unsustainable *Motherboard* 2015-06-29
4 Peter Fairley The Ridiculous Amount of Energy It Takes to Run Bitcoin *IEEE Spectrum* 2017-09-28
5 Cardano Ouroboros

The blockchain selection mechanism selects a single blockchain when several alternative chains exist. The blockchain database may differ between the participants of the project. Due to the latency of network propagation or temporary splits in the network, the database may have incidental forks. Accidental forks also happen, such as during software upgrades where the new code accepts different data (transactions or blocks) from the existing software. Intentional forks also happen, sometimes during software upgrades of the network, at other times when a blockchain project splits into two separate projects, each to follow its own chain. Blockchain projects include a mechanism to select one chain as authoritative. Bitcoin adopted the *longest-chain* rule which selects the valid chain as the one with the most blocks. While two chains are the same length, it is unclear which will become authoritative; only the next block, building on one chain or the other, will determine which is longest. Since Bitcoin uses a proof-of-work block generation mechanism, the longest chain rule is equivalent to selecting the chain with the most invested work.

This resolution mechanism introduces a known vulnerability to the system. An actor with control over a majority of the network's processing power can take control the network simply by working faster than the rest of the network. Since the majority of processing will generate a chain which becomes longer than that generated by the rest of the network, the chain generated by the actor with control will become the official record. This is termed a 51% attack. This weakness was recognized as early as in the original research paper presenting Bitcoin but it was hoped that the attack would not be practical and that selfish economic incentives would prevent actors from undermining the cryptocurrency. However, the explosion in alternative currencies have led to many such attacks[1] and the ability to rent machines for processing has made the cost of the attack quantifiable[2].

Networks

All blockchain projects are necessarily supported by a network of computers. The networks enable the computer nodes to share the blocks in the blockchain database and to share all requests for modification of the chain through transactions.

The physical network used by blockchain projects is generally always the TCP/IP based Internet. This is the most readily available, global scale networking infrastructure. Therefore, the *network* discussed here is the virtual network comprised of all the nodes participating in the blockchain project.

The networks used by blockchain projects range from homogeneous networks in which all computers are equivalent and run the same code, to highly differentiated networks in which different computers perform different tasks. The structure and complexity of the networks, the roles played by each type of computer, the code run by the different types of computers, and the messages exchanged between computers all serve to distinguish between blockchain projects.

Here we consider three separate types of networks. Homogeneous networks consisting of identical, full nodes and without any restriction on participation are the simplest networks. In such networks, the nodes must each perform all of the work needed to maintain the network and the computation. The Bitcoin project started with such a homogeneous network. Heterogeneous networks include nodes of different kinds, for instance with nodes which provide access to the network to outsiders or with nodes which provide access to pools of nodes with specialized hardware for mining calculations. This is the structure of the Bitcoin network today. Permissioned networks are newer

1 Tyson O'Ham PoW Not Flawed Regardless of Rash of 51% Attacks in Crypto *Bitsonline* 2018-05-25
2 51crypto

and even more heterogeneous; they generally include separate nodes of the network which are uniquely responsible for access control.

Communication in a blockchain network involves the exchange, through various mechanisms, of multiple types of messages, between the various computers. The specific mechanism for message exchange provides one characteristic of the project. Some projects use simple request-response type exchanges, possibly repeated in a polling pattern. Other projects include a subscribe and publish (*pub-sub*) style of exchange. Some projects adopt a specific message bus subsystem for communication. Some of the communication may actually take place through the blockchain itself. For example, the votes of miners for the evolution of the software of some projects can be cast through comments directly within transactions of a certain type (coinbase). Also, blockchain projects which support distributed applications (Dapps) may provide a means for them to communicate between themselves, such as the "whisper" mechanism used by Ethereum.

Homogeneous

The simplest blockchain projects rely on a network in which all of the computers acting as nodes run identical code and perform identical functions. These networks are almost all open (permissionless) meaning that any new participant can start a new node and join in the work.

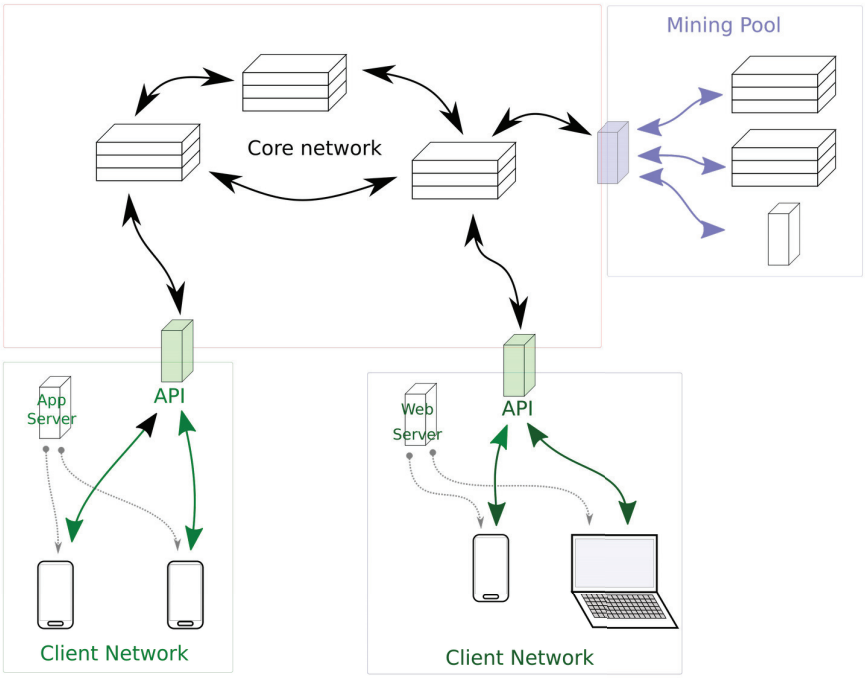
These nodes of a homogeneous network are responsible for maintaining connectivity between nodes, sharing communication messages between known nodes, maintaining a copy of the blocks of the chain and of the pending transactions, validating those data structures, participating in the extension of the database, and enabling the use of the system. The Bitcoin blockchain project started as a homogeneous network of this kind.

Heterogenous

More realistic blockchain projects use networks comprised of heterogeneous nodes, specialized for particular tasks. For example, larger networks that allow broad public participation generally have nodes specialized to handle those users without requiring such users to run a node continuously on the network.

The Bitcoin project, as it grew in size, changed from a homogeneous network to a heterogeneous network. The increased cost and economic incentives of mining drove some participants to combine their efforts into mining pools, where one set of machines prepared a mining template while other machines, sometimes with specialized hardware, actually performed the work of mining. Similarly, the same growth led to the differentiation of nodes intended to support users. The majority of users run lightweight client software, usually called *wallets*, that do not participate directly in the network but rely on some trusted party to provide access to the blockchain service.

A modern blockchain network, in a generic view, consists of a core, which does most of the processing, of gateways, both to mining pools and to clients, and of a periphery that handles the clients.



Computers in the core network handle the communication, processing, and storage needed to maintain and extend the shared ledger. These machines will generally each have a copy of the entire blockchain. These computers will have processed the blockchain to derive and validate the current state of the system. These machines must be able to communicate with other machines in the core so as to share the blocks of the chain, accept and share transaction requests, possibly communicate to reach consensus on a new block, and then distribute the new block throughout the network. These machines are also responsible for validating new transaction requests prior to processing them and validating the new blocks when published on the network. Specific blockchain projects will differ in the organization of this core network, in the code used by the computers on this network, by the messages exchanged, and by the procedures through which a new block is created.

Computers in mining pools, present especially in blockchain projects which use a proof-of-work consensus mechanism which allocates block rewards and pays fees, collaborate to solve the cryptographic challenge and thereby generate a new block to add to the blockchain. These computers generally run code especially built for the computer hardware. Generally, one set of machines, acting as the gateway to the core network, gather up pending transaction requests into a candidate block. The violet coloured node in the image above is such a gateway. These machines, or others, orchestrate the pool by distributing the candidate block template to be mined, by receiving partial results to show the ongoing contribution of each machine to the effort, by accepting any valid block found, submitting that result to the core network, and distributing the rewards to the participants in the pool according to the amount of intermediate contributions made. The software and messages used in these pools is independent of that of the core network.

Computers which provide gateways between the core network and the clients generally provide both software and services. The green coloured nodes in the image above are client gateways. The software provided by the gateway systems to be run by clients, generally either as web pages or as mobile apps, tends to be lightweight, without handling the blockchain directly. The services provided by the gateway systems may translate between the programming language of the core network and the language of the clients by providing a formal application programming interface

(API), often in Javascript as used in web pages. The messages exchanged between these gateway systems and clients are generally completely independent of the core blockchain system. Computers used by clients usually run lightweight code which interacts with the core network through special gateways. The resource requirements of participating in the core network, in terms of processing power, memory, bandwidth and storage, are usually too high for everyday computers such as mobile phones and laptops. Instead, clients run a so-called *wallet*[1] which enables clients to submit new transaction requests to the core network. Note that this requires the clients to trust the gateway. Gateways may also simply provide information to users about the blockchain, such as offering a query or search interface to obtain information on the current state of the blockchain[2,3].

Permissioned

Recent blockchain projects, such as the permissioned blockchains, use networks with heterogeneous nodes. The nodes handling the permission system are generally dedicated solely to that purpose and so differ from the transaction and blockchain processing nodes.

Permissioned blockchains can control the access of participants to the system or can control their use of the system. Usually, permissioned networks restrict access by default. In order to use the network, users may have to authenticate and obtain credentials. Similarly, in order to participate in the network, operators may have to authenticate and obtain credentials for their nodes. Access control usually takes the form of digital certificates and signatures based on public-key cryptography. Permissioned networks can also control the use of the system. In order to use certain parts of the network, the network may need to be configured to allow that use.

Access control to digital networks generally follows a standard public key infrastructure model in which digital certificates, signed by certificate authorities, vouch for the participant. Parties wanting to use, or participate in, the system first authenticate to some system and obtain a digital cryptographic certificate; each use of the system then requires signatures based on the certificate. The original Bitcoin system managed to avoid access control: access control ran contrary to the goal of forming a global system for universal payment, access control would have complicated the security analysis of the cryptocurrency, and access control, by introducing a mechanism of exclusion, would have introduced another point of potential abuse. Current blockchain projects which are developed for a specific community often include access control both to eliminate any potential abuse by outside users and to provide an audit trail for the system.

Policy control in digital networks offers one approach to control usage of the system. Policy control requires the definition of a set of policies and implementation of points of control where the use requested of the system is compared against the use allowed by the policies. Robust and flexible policy control systems may assign participants *roles* and write policies against those roles, thereby separating the definition of what use is allowed from the determination of who is allowed that use. The Bitcoin project did not control use of its system, thereby keeping the system simple. Newer blockchain projects, such as those built with Hyperledger Fabric, can include complex policy rules for the submission, validation, and acceptance of transaction requests.

1 This nomenclature is unfortunate because, unlike an actual wallet, this software generally does not actual hold any data structure of value but rather holds the cryptographic keys necessary to control value held within the blockchain itself. These are keychains and search tools more than they are wallets.
2 Blockchain.com
3 Etherscan.io

Permissioned blockchain projects can implement access control and policy control. Even more complex network architectures are possible, such as the system supported by the cloud computing service provider Azure which can handle multiple underlying blockchain systems[1].

Machines

All blockchain projects requires computers to run, or take part in, the system. These machines range from general purpose, personal computers to specialized machines running in data centers. The ability of potential participants to obtain, operate, and maintain computers which can participate in the blockchain calculation deeply affect the character of that project.

The democratic intent of the original Bitcoin project meant that its original design considered a network of general purpose, personal computers. These computers are widely available and relatively affordable so that the requirement for such machines did not limit participation significantly. (The requirement to have access to the Internet was the most fundamental limit to participation.) Personal computers have enough processing power, memory, disk storage and network bandwidth to perform all of the functions required of a node on a blockchain network.

The use of more powerful machines arose first in the process of block generation, also known as mining. Blockchain projects which allocate the right to generate a block and collect its rewards to the first node to solve a computationally difficult problem, underwent a process of machine specialization. The immense growth in value of the cryptocurrency meant a parallel increase in gains to be made from mining; this lead to massive investments in the computers taking part. The first steps in specialization were to move the mining computations from the general purpose central processing unit in every computer to the graphics chips, leading to massive demand for the most powerful graphics cards on the market. This step, while it increased the expense of participating in mining was generally available to all. However, specialization continued, first to use field programmable gate arrays (FPGA) chips which could be specially configured to speed up the mining calculations. The final step in specialization has been to create application specific integrated circutics (ASICs) which are chips specifically designed and produced to speed up the mining calculations. This latter step is only available to participants able to invest heavily. As the costs of participation rise and the difficulty of obtaining suitable hardware increase, the networks become less democratic and participation becomes limited to only deep pocketed, specialized participants.

Software

Blockchain projects require computer software code to run. Each of the nodes running on the network require code for communication, validation, storage, and processing. Clients of the network need code to present an interface to the user and to communicate user actions to the network. User actions generally involve script code which can be arbitrarily complex in some blockchain projects.

Building this code, maintaining it, and extending it requires even more code such as integrated development environments. Similarly running the network requires monitoring and supervision code.

1 Microsoft Azure Blockchain Workbench Architecture 2018-04-20

Node code

The main code generated by a blockchain project is the software used to run the nodes on the network.

This *node code* differs in each blockchain project due to the different functionality offered by each blockchain project. However, full nodes generally include a common set of functions: communication code to maintain the network, storage code to archive the blocks of the blockchain database and to store pending transactions, validation code to check the blockchain and pending transactions, and possibly block processing code to generate new blocks. The client code to actually use the system is logically independent of the code of the node and often provided separately.

Communication code is required for all nodes in any blockchain project since the nodes of the networks need to exchange messages including the blocks of the blockchain database and the transactions submitted by clients but not yet processed into the blockchain. The Bitcoin communication protocol include periodically announcing the node's IP address to other peers, rebroadcasting transactions newly announced by neighbouring peers, and sending heartbeat messages[1]. The Ethereum project developed its own peer-to-peer communication protocol[2,3].

Storage code is required for any node which acts as a relay or validator. Relay nodes need to store temporarily recent data structures such as new blocks and pending transactions so as to be able to re-transmit those to other peers. Validation nodes need to store the entire blockchain and possibly the currently calculated state against which to validate new blocks and transaction requests. The actual storage may be directly onto a file system but more commonly uses a key-value store backend such as LevelDB[4] or CouchDB[5].

Validation code uses validation rules to establish the minimal acceptability of the data structures in a blockchain project. The validation rules for the shared blockchain data structure are generally hierarchical. A chain is valid if all its blocks are valid. A block is valid if the hash which it records for the proceeding block matches the value obtained when a hash is calculated on the previous block (or its header), if the block contents are valid, and, optionally, when other conditions are met. Block contents are valid if all content elements are valid while the rules for the validity of content elements are particular to each blockchain project. Project procedures generally call for all nodes to establish the validity of the blockchain prior to performing any other actions, although enforcing that rule is often impossible.

Processing code creates new blocks by assembling pending transactions into an ordered list of transaction results. In the Bitcoin blockchain project, processing is done by miners individually or in pools. Bitcoin miners assemble transactions into a set, validate each transaction in turn to ensure that the inputs are not yet spent, and generate a block's contents with those transactions. The miner then performs an extensive search for some combination of these transactions and a nonce value which leads to a block whose hash value has the characteristics required by the proof-of-work mechanism. The Hyperledger Fabric project splits its processing code between two heterogeneous node types, peers and orderers, in an execute-order-validate flow. First, clients send a transaction request to one or more peers. The peer evaluates the transaction and, if the peer accepts the

1 Bitcoin Network Retrieved 2018-07-31
2 Ethereum ÐΞVp2p Wire Protocol Retrieved 2018-07-31
3 Ethereum Ethereum Wire Protocol Retrieved 2018-07-31
4 Leveldb.org
5 Apache CouchDB

transaction, the peer calculates the result of the transaction and endorses the request and result with a digital signature. Once sufficent endorsements are collected, the pool of endorsed transactions is sent to an order node which merely establishes some order to each of the transaction sets and broadcasts the blocks to the network. All the peers on the network then validate or invalidate each transaction set and use the valid transaction to update their calculations of the current state of the system.

Processing code includes the *consensus algorithms* which define the method through which participants in a blockchain project reach agreement both on the current state of the system and on subsequent changes. These algorithms are among the most discussed components of blockchain projects since consensus lies at the heart of the trust shared by all participants in the system, though not in each other, and because consensus must be reached in the face of incomplete synchronization or even active attack. Two aspects of consensus can be considered: consensus on the past history of state changes and consensus on how to generate a new set of state changes. Agreement on the past centers around the principle of the longest (valid) chain: all participants are expected to use the longest chain of blocks of which the participants are aware. Agreement on the next step requires agreeing on a set of transactions to be included, agreeing on the order, agreeing on how the block will be generated and who will receive any rewards, either new assets or fees, permitted by the system.

Several methods of generating the next block have been proposed. The Bitcoin project adopted the proof-of-work consensus mechanism in which multiple participants race to solve the computational puzzle of finding a block for which the hash value of the header is below a given threshold. The block eventually accepted into the blockchain of record assigns to its finder the reward for the block and the fees for processing the transactions. Because the proof-of-work mechanism is intentionally inefficient, other consensus methods have been used. The Ripple project uses a method they call 'consensus' in which multiple participants agree on the set of transactions to include. Several projects use a consensus mechanism in which one participant, from a pre-defined set, is elected to handle the transactions during a given time slice. Several other projects use a proof-of-stake method in which participants risk a stake in order to be part of the consensus pool. Consensus mechanisms based on other mechanisms include proof-of-elapsed time, and several others[1]. The cryptographic guarantees and game theoretic economic incentives of the different consensus mechanisms lead to different trust guarantees in the result.

Wallet code

Client code allows participants in a blockchain project to actually use the code. This code provides a user interface to show the current state of the system (at least the parts of that state of interest to the user) and to create new transactions to change that state. For instance, in a simple cryptocurrency network, the client would present the user with the funds available in the various user accounts. The client code also provides a way to send transaction requests out to the network.

Client code is usually called a software *wallet* in a cute but inaccurate parallel to the physical world. Where a physical wallet actually holds a user's cash, a software wallet merely provides access to the resources, usually by managing the private cryptographic keys with which users can sign any transaction spending the resources held in the accounts.

1 Seibold Sigrid and George Samman *Consensus: Immutable agreement for the Internet of Value* KPMG 2016

Confusingly, client code comes in many varieties. Some clients can interact directly with the blockchain projects. Other clients, such as those provided by exchanges, provide an indirect connection first through the servers of the exchange and then to the blockchain.

Chain code

Blockchain projects which allow transactions with complex procedures introduce an entire new realm of software code. This code is often called a 'smart contract' which is confusing since the code is not particularly smart nor does it necessarily act as a legally binding agreement on the level of a business contract. Here, we will call such code *chain code* since it is software which lives on the blockchain and which is executed by the blockchain system as a whole.

Chain code introduces a whole new level of complexity for software developers: in the languages used, in the environment in which the code executes, and in the security issues involved.

The software language in which chain code must be written can be specific to each blockchain project, although newer projects attempt to allow more mainstream languages. The limited scripting system provided by the Bitcoin project uses a language unimaginatively called 'script' which is intentionally limited to ensure all scripts run to completion and not freeze the computation. The Ethereum project's central focus was on extending scripting to allow for a Turing complete[1] language. To ensure all scripts would run to completion, the Ethereum project incorporated a cost of computation which had to be paid thereby ensuring that the computation would either finish or run out of funds. Ethereum developed a language called Solidity[2,3] for its chain code but, since the code is complied to byte code, other languages can also be used. Newer blockchain projects, such as Hyperledger Fabric, allow the use of common programming languages such as Javascript and Go.

The execution environment of chain code differs between blockchain projects. Most code executes within virtual machines whose characteristics are defined by each project. The execution environment usually provides access to some contextual information such as the current block number for the blockchain. The execution environment may provide access to permanent storage and may allow sending messages or making remote procedure calls to other chain code in the system.

The security issues involved in chain code execution are extensive and remain incompletely understood. Vulnerabilities are common and sometimes spectacular: the first attempt to form an organization for investment on the Ethereum blockchain collapsed due to a security failure in the code. Chain code is generally published on the blockchain, transparent to all giving attackers plenty of time to search for vulnerabilities and to exploit them.

While chain code opened up a vast realm of innovation, it has also opened up a whole new world of complexity and vulnerability. Resolving these will be the task of a next wave of work.

1 The concept of Turing completeness defines the maximum threshold of computational complexity of current computers.
2 Ethereum The Solidity Contract-Oriented Programming Language
3 Wikipedia Solidity Retrieved 2018-07-31

People

Blockchain projects, as collaborative computations, depend on the people in the community. People individually must decide to participate in a project, and then must collaborate to build, run, and use the system. Blockchain projects rely on cohesion of its participants but current projects lack effective mechanisms of self-governance through which to maintain communal cohesion. This has resulted in many splits in the communities of different project and in the search for effective mechanisms of project self-governance.

Blockchain projects require a community of participants to administer the project, to develop and maintain the software code, to operate the network and machines which perform the distributed calculation, and to use the system. In the first blockchain systems, most of the work of maintaining the blockchain system itself fell to unpaid volunteers, contributing in the tradition of free software development. These volunteers were responsible for registering the domain, building and updating the website, managing the discussion systems (email lists, web fora, or topics on Reddit[1]), writing the software code, operating the default, initial nodes from which clients could bootstrap, and possibly owning the trademarks. The only financial incentive included in the systems was for the operators of the network who were given a reward with every block generated.

Blockchain projects therefore depend on community cohesion but face many dispersive forces. For transparency, the source code for most blockchain projects is published online; however, this enables anyone to create a copy of the project rather than participating in the original project. Also, a running network can easily be split into two, if part of the community so chooses. (Indeed, such splits are actually the way most software improvements are adopted by the blockchain networks.) Maintaining a cohesive set of goals of the project is difficult as circumstances change, as experience shows what works, as new ideas are proposed, and as a massive wave of new participants joins the project.

This dependency of blockchain projects on the cohesion its community undermines the trust model that the first blockchain project, Bitcoin, sought to establish. The original goal of these systems was to avoid trust in any specific third party by building a system transparent in its operation and sufficiently distributed in its execution to ensure no single party could control the system. This would allow trust in a third party to be exchanged for trust in the blockchain system. Unfortunately, the reality is that one also needs to trust the community itself, which is a single, centralized entity, to update the system as problems arise. Thus, the blockchain projects fail to eliminate trust in a third party and to fully decentralize the system.

Blockchain projects do no have effective mechanisms to maintain community cohesion. The only formal mechanism in common use is a financial reward given to those who generate new blocks of the shared data archive. The informal mechanisms of cohesion are not especially effective. Charismatic leaders often provide for project cohesion but the very first project, Bitcoin, had its only obvious leader disappear into anonymity. In reality, blockchain projects adopted a very weak form of cohesion. Project administration was mostly done by those who first established the project. Project development followed the pattern of free software projects in general with a core set of committers able to change the code, and a semi-formal proposal process used to build consensus for

1 Reddit.com

significant changes. The informal nature of project leadership, fueled in part by the desire to be decentralized, led to a system which is not formally accountable in any way.

This lack of formal governance has had several consequences, not the least of which is vehement debate over even the slightest of changes to the blockchain system. The developers of the original Bitcoin project became increasingly conservative in their approach, allowing less and less change to the system, inhibiting response to changing circumstances. Many projects have undergone complete splits of the community. Bitcoin has split multiple times, mostly over how to enable the system to scale to handle many more transactions: Bitcoin XT forked in 2014, Bitcoin Classic in 2016, Bitcoin Cash in 2017, each wanting to increase the block size, while Bitcoin Gold forked in 2017 to adopt a new consensus mechanism aiming to reduce concentration in mining. Similarly, the Ethereum project has split, with the Ethereum Classic fork maintaining the original rules of the project while the core Ethereum project introduced new rules to revert the transactions of a failed smart contract. When disagreements do occur, such as in the case of Ethereum, they lead to profound crises in the community and negative feelings for the loosing side.

The response to these issues has been two fold: on one hand, to study and discuss the issues, and, on the other hand, to explicitly tackle the issues. A number of academic researchers, such as Primavera De Filippi[1], study the self-governance issues of blockchain projects[2,3]. Primavera De Filippi speaks of "the tyranny of structurelessness"[4,5] (echoing "the tyranny of the majority") being a defining trait of the informal governance of these communities. Since the communities are purely voluntary, participants can choose their own response to every single decision. Since leaders are not formally established, they are not accountable. Similarly, members of the community have been publishing about the issues[6,7,8]. Newer blockchain projects are addressing community self-governance directly. The Tezos blockchain project aims to place the governance rules directly on the blockchain and automate the introduction of changes through a formal voting mechanism[9]. The Cardano project also aims to address governance issues formally[10]: first by building a foundation and then by building a trust to collect funds and to spend them where needed.

1 Primavera De Filippi Governance by Design - Primavera De Filippi - OuiShare Labs Camp #3 YouTube 2015-12-16
2 Primavera De Filippi and Benjamin Loveluck The invisible politics of Bitcoin: governance crisis of a decentralized infrastructure Internet Policy Review 5(3) 2016-09-30
3 Mike Orcutt Leaderless Bitcoin Struggles to Make Its Most Crucial Decision MIT Technology Review 2015-05-19
4 Primavera De Filippi and Aaron Wright Blockchain and the Law: The Rule of Code (@1:03:44) YouTube 2018-04-27
5 Jo Freeman The Tyranny of Structurelessness 1970~1973
6 Fred Ehrsam Blockchain Governance: Programming our future 2017-11-27
7 Vlad Zamfir Against on-chain governance: Refuting (and rebuking) Fred Ehrsam's governance blog 2017-11-30
8 Vitalik Buterin Notes on Blockchain Governance 2017-12-17
9 Tezos On-Chain Governance 2018-07-28
10 IOHK IOHK | Cardano whiteboard; overview with Charles Hoskinson 2017-10-26

Blockchain, the Information System

In our review of the world of blockchains, we can focus on the blockchain as an information system, that is a mechanism of people, procedures, and computations which processes information to serve a particular purpose.

Blockchain projects, that is the communities, networks, and technology involved in maintaining an ongoing computation to extend a shared blockchain database, have been proposed for many types of applications. Proposed uses include timestamping services, cryptocurrency services, registries, election systems, and supply-chain management, although new uses are being proposed and invented every day. Some of these proposals have been realized as running systems, other proposals are in development, while others are still being designed, or even conceptualized.

There are many reasons *not* to use blockchain data structures or run blockchain computations.

Blockchain projects are inefficient compared to centralized relational databases, blockchain projects require coordination among multiple participants, and blockchain projects do not scale as readily as other systems. Blockchain projects only make sense in a limited number of situations where multiple separate parties modify system state with no central coordination or mutual trust. Blockchain projects have many vulnerabilities: for example, the community running the project could change the rules of operation or the integrity of the blockchain could be destroyed by a hostile majority of operators running the computation.

Several publications present decision flow diagrams to help project designers assess the relevance of a blockchain for their projects, from a joke decision tree with the constant answer 'no' to more complex models[1]. A widely cited paper[2] presents a decision flow in which most paths reject the use of blockchains but suggest a blockchain based design could be of use for projects which simultaneously need a shared record of events, involve many, mutually untrusting, participants who can each change the state of the system, and have no intermediary trusted by all.

While these decision flows are generally correct in recommending project designers look with caution at blockchain based designs, these recommendations fail to stress one of the most important contributions of blockchains: founding trust in a distributed community through a shared database of all changes. Blockchains are not the only possible shared database but are one solution. Using blockchains as the digital foundation of public trust can reinvent the relation between the public and government institutions. Parliaments can publish a full record of all activities, ministries can publish a complete record of all decisions, and judiciaries can publish a record of all judgements. Blockchains can provide the digital equivalent of the official public publications which governments use to declare legal changes, such as the publications of the Office of the Federal Register in the United States or of the *Dirección Nacional de Impresiones y Publicaciones Oficales* (IMPO) in Uruguay. Blockchains can also form the public record for national registries such as the cadastre, property ownership, and civil status. Blockchains could also play a role in the official record of private entities such as registering the decisions of a board of directors or registering all changes to official data sets. Even when the use of blockchains might otherwise be

1 Sebastien Meunier When do you need a blockchain? Decision models. *Medium* 2016/2017/2018
2 Karl Wüst and Arthur Gervais Do you need a Blockchain? *Cryptology ePrint Archive* 2017-04-27

disadvantageous, their value as immutable archives to be shared as official records might argue for their use.

Here we consider several areas of application for blockchain systems. These application types are grouped by the core nature of the information system design: systems which make simple declarations of fact, systems which track assets, systems which invoke previously stored procedures, or systems which interact with multiple blockchains at once. Obviously a single project can support several of these information system types at once; however, for clarity in this presentation, we consider each type of application separately. We consider here the fundamental nature of the information systems implemented using blockchain projects not the details of the implementation. Different implementations might provide for a similar flow of information while differing in who can participate in the system, how participants are identified, or what participants are authorized to do.

Declaratory Systems

The use of blockchain projects to record declarations into a blockchain database forms one of the most important applications of the technology, despite being frequently overlooked. Declaratory blockchain systems can serve as a trusted log of events, as testimony to the existence of certain resources at certain times, or as archival storage.

The mechanism for simple declarations has evolved in time. Originally, in the Bitcoin blockchain, declarations were introduced in unused fields of transaction requests. Thus a transaction's signature script field could be reused for arbitrary content; the script would fail to validate the transaction so the transaction would have no effect but the data would become part of the blockchain data archive. This approach was eventually formalized. This usage is important enough that modern blockchain projects generally include an explicit mechanism to allow users to make arbitrary declarations.

Declarations form an important component of blockchain systems, although this was not recognized originally. Ironically, Bitcoin made use of such a declaration before even creating the first coins: the coinbase transaction of the genesis block contained a newspaper headline both as a political statement and as an external timestamp preceding the origin of the system. While the use of cryptographic hashes for time stamping documents had motivated one of the lines of research which led to blockchain systems, the value of these arbitrary declarations in the Bitcoin system was only discovered in time. The first significant use for these declarations was to "overlay" of external systems, such as was done by the Counterparty project. The need for declarations, and the ad-hoc approaches used to make them on the Bitcoin blockchain, eventually led Bitcoin developers to include the `OP_RETURN` instruction for the virtual machine. The instruction would end processing of the script, rejecting its instructions, but allowed up to 40 bytes of arbitrary content to be stored on the blockchain.

Since the information recorded is arbitrary and possibly incomprehensible to the blockchain system recording it, the use of declarations usually requires that the blockchain archive be processed through external software.

Blockchain systems which allow the recording of declarations into the shared, archival database have many potential uses. These projects can provide official histories of events or formal presentations of announcements.

Log Book (records)

The use of a blockchain project as a recording system to provide an authoritative record of public events provides a fundamentally important application of blockchain systems. In this role, the blockchain project acts as a digital log, with the cryptographic elements ensuring immutability of the record. While this usage seems trivial, it has numerous applications in government, private entities, and civil society.

While blockchain systems can provide simple record keeping, since blockchain systems automatically associate records with the time at which they are recorded, most uses will rely on dated, or time stamped, records.

The overlay concept involves one project recording its events into a separate blockchain system. The blockchain system merely acts as a recording engine, including records in the blockchain archive. The project writes special software to interpret the blockchain archive according to the project's rules. The project thereby can make use of the blockchain system's distributed network, trust, and replicated database archive for a purpose unrelated to that of the blockchain system itself.

Timestamping (dated records)

The use of a blockchain project as a timestamping service provides another valuable application of the technology. A trusted timestamp[1] provides proof that a certain digital resource existed at a certain moment, thus enabling scientists, inventors, authors, or other creative professionals to establish the existence and priority of their work. The search for a design for timestamping services actually provided some of the antecedent research which led to the invention of blockchain systems[2], in the work of Stuart Haber and Scott Stornetta[3] among others. Note that for trusted timestamping we can not rely solely on the blockchain system since this could lie; instead, the blockchain system must be anchored to some external source of time such as another, more trusted, blockchain or a newspaper of record.

Timestamping can be performed relatively trivially from an arbitrary digital resource. A cryptographic hash function is selected and used to calculate, from the digital resource, a hash value. The hash value is then included in a declaratory transaction to be embedded in the blockchain. The recorded hash is timestamped by the time declared for the block (along with the times declared in a few subsequent blocks to allow for differences in the clocks of the various computers generating blocks in the network[4]). At any future time, the existence of the resource can be proved by providing the resource, recalculating the hash value using the same approach as done originally, and showing that the same hash value had been recorded in the blockchain at a given time. As long as the blockchain system's blocks are trusted to have occurred at the time they record

1 Wikipedia Trusted Timestamping Retrieved 2018-07-17
2 Arvind Narayanan and Jeremy Clark *Bitcoin's Academic Pedigree: The concept of cryptocurrencies is built from forgotten ideas in research literature* ACMQueue 2017
3 Stuart Haber and W. Scott Stornetta How to Time-Stamp a Digital Document Advances in Cryptology-CRYPTO '90 3(2):99-111.
4 The timestamps of each block are subject to the time skew naturally present in distributed networks due to either bad configuration of certain servers, thermal drift of the oscillators, or other factors. Even properly configured servers can differ by several milliseconds, although the new Huygens protocol lowers that to nanoseconds.

in the blocks (either because the blockchain is widely followed or due to some external mechanism such as the regular publication of block hashes in a printed newspaper), this provides sufficient proof that the digital resource existed prior to the time given by the block. Note that the digital resource itself is generally not recorded in the blockchain since the use of a hash offers the same guarantees without using as much storage space.

Timestamping is especially effective using a blockchain system which has a high volume of unrelated transactions since that traffic serves as an extra guarantee of the integrity of the blockchain system. The use of the Bitcoin or Ethereum projects for timestamping is especially attractive since no one can reasonably worry that those high-volume blockchains could be falsified later. Several projects currently enable timestamping[1] on the Bitcoin blockchain system, although it is relatively simple for a user to build one's own system.

Blockchain systems which archive digital records, relying either on mere existence of the record, on the order of the records, or on the specific times at which the records were made, can serve multiple purposes.

Governmental entities would do well to adopt blockchain record systems as the digital form for the official public record. Parliaments could record all of their actions as blockchain records including parliamentary procedures, votes and other decisions, and even the official transcript of parliamentary sessions. Executive branches could record meetings, discussions, and new regulatory declarations. Judiciaries could record all judgments and opinions as well as documenting the events in public trials.

Private entities such as public companies could also adopt blockchain record systems as the digital form for their official records. The agendas, minutes, and decisions of the board of directors could be officially recorded on blockchains, as could the decisions of the executive officers.

Civil society could also make use of blockchain record systems, especially in situations with multiple participants. The modern, digital form of the stock market ticker tape could transition to a blockchain based system. Similarly, official betting pools could announce all bets on a blockchain so as to publicly demonstrate the inputs to the odds calculations for their parimutuel system.

Individual researchers or inventors can use blockchain systems to document their research in a digital analogue to the paper lab book both to establish the steps in their research and to establish their claims to priority.

Voting

The use of a blockchain system as part of a voting process has natural appeal. Conceptually, votes could somehow be registered on the blockchain, whose immutability guarantees would make it a trusted repository of all votes for subsequent review. The transparency of a blockchain used as a shared ledger of events could enable multiple parties to calculate the results of the elections and leave little room for argument. Unfortunately, the transition from this insight to real world use runs into many difficulties, as explained below.

1 For example, Origin Stamp, Open Timestamps, or Woleet

The concept of voting systems which could be audited from end-to-end[1] has been explored by many researchers at least since the work of David Chaum[2,3]. The most convincing systems use a combination of paper ballots, to enable trusted recounts, printers, to avoid user errors, and cryptography, to provide the proof that a particular vote has been included in the final count without allowing any demonstration of how the vote was cast.

However, this research, while exciting and valuable, has not yet resulted in a persuasive solution for elections. The designs may not be generally applicable. The design of these approaches has been dominated by work in the context of elections in the United States of America where voteing tends to be for individuals and includes multiple elections at once; it is not clear how such approaches could be applied to other voting systems such as systems voting for political parties. The designs may only apply to part of the chain of trust in elections. Trust in voting systems rests on several facets including preparatory phases where eligible voters are registered and ballots are prepared, execution phases where eligible voters cast their votes, and the tallying phase where votes are counted. Trust relies on the secrecy of the actual act of voting to ensure the individual can vote free of coercion: voters must be able to deny plausibly how they voted and must be unable to demonstrate in any way how they voted. The fatal flaw of these designs might be in their very reliance on cryptography: trust in any voting system requires that the system be transparent to all but the mathematics of cryptography makes these digital systems difficult to grasp.

While the use of blockchain systems in elections have been proposed and several projects are actively attempting to apply blockchain systems to elections, none of these offer a credible, effective solution to real world elections.

The Agora[4] blockchain project aims to provide a blockchain based voting platform for the digital world. However, the system does not appear to ever have been successfully applied. Futhermore, the press releases[5] surrounding an observation mission to the elections of March 2018 in Sierra Leone, led to a large controversy[6] and blacklash[7] against the project, requiring an official response[8]. That response showed that this experimental application of the Agora system only came into play after voting, when the results of counts of each separate ballot box was entered into the blockchain system. While this could be a valuable contribution, it falls short of the end-to-end goal. Agora is mostly proposed as a system to allow voting using digital devices (that is, without physically going to a polling location) but, since this does not solve the issue of voter coercion, such solutions are unsuitable to use in national elections.

The Democracy Earth Foundation[9] also proposes the use of blockchain systems in elections, although with an even greater ambition of enabling ongoing governance without permanent representatives. The concept of liquid democracy proposes that electors could vote directly on some issues and delegate their votes to representatives on other issues. The Democracy Earth project hopes to build a system in which authenticated identities combine with vote tokens and a delegation

1 Wikipedia End-to-end Auditable Voting Systems Retrieved 2018-06-15
2 David Chaum Secret-Ballot Receipts: True Voter-Verifiable Elections IEEE Security and Privacy 2(1):38-47, 2004
3 Ron Rivest has a collection of research up to 2004.
4 Agora.vote
5 https://www.agora.vote/press
6 Kristin Houser Hold Up: What Actually Happened in Sierra Leone's "Blockchain" Election? *Futurism* 2018-04-02
7 Shawn Gordon Sierra Leone and teh Blockchain Election that Wasn't *Bitcoin Magazine* 2018-03-22
8 Agora Research Team Agora Official Statement Regarting Sierra Leone Election *Medium* 2018-03-19.
9 Democracy Earth Foundation

system to enable participants to vote directly with their tokens or pass their tokens temporarily to a delegate who can vote on the participant's behalf. As of now, the system does not yet exist but there are plans for pilot projects in the near future[10].

The lack of any successful examples coupled with the inherent difficulties of translating voting from the physical world to the digital suggests that this area of application of blockchains, while of great interest, remains a hope for the future rather than a reality of the present.

Asset Tracking Systems

The use of blockchain projects to record the existence, ownership, transfer, and withdrawal of *assets* provides an exiting application of the technology.

The mechanism for tracking assets on blockchains has evolved. The original Bitcoin project only handled transactions related to the single Bitcoin cryptocurrency. Two strategies were explored initially to expand the Bitcoin system to handle other assets: overlays and colored coins. In the overlay strategy, the Bitcoin blockchain was used only to store the transaction records of the overlay system while separate software tracked the ownership of assets in the overlay system. In the colored coin strategy, a coin (or fraction) was selected as a representation of another asset and the Bitcoin blockchain was used to trade this asset. The arrival of second generation blockchains with their stored procedures offered a much more complete solution for asset tracking, as is discussed below.

Asset tracking systems can be split into two classes: asset systems which track digital assets defined within the blockchain itself (*on-chain* assets), such as a cryptocurrency token or domain name identifier, and asset systems which track real world assets (*off-chain* assets) represented through a digital asset, such as a piece of land or a deposit of fiat currency. In the latter case, the blockchain system acts as a *registry* for the real world asset, using the ownership of the digital representation to signify ownership of the real world asset. While the tracking of ownership of on-chain assets relies only on trust in the blockchain project itself, the tracking of ownership of off-chain assets necessarily places trust in some third party actor, the one who defines the relation between the real world asset and its representation within the blockchain system.

Asset tracking systems involve three operation types: asset creation or registration, asset re-assignment, and asset destruction or deregistration. For example, the Bitcoin on-chain asset system creates coins through so-called *coinbase* transactions added to each block by miners as their own reward and transfers coins through regular transactions which assign ownership to some other identity. The Bitcoin system does not have any formal method to destroy coins (although this can be done informally by transferring the coins in such a way that they can, demonstrably, never be used again).

Cryptocurrency

The use of a blockchain asset tracking system to define and exchange on-chain, provably scarce, digital elements as representations of value with no external backing, that is for a cryptocurrency, has been the central use of blockchain technology to date. The success of these cryptocurrencies has been a driving motivator in the growing interest in blockchain systems.

10 Democracy Earth Democracy Earth Development Update Apri-May 2018.

This discussion is <u>not</u> an endorsement of cryptocurrencies.

This discussion is a *technical presentation* of information systems which use blockchains for the ownership and exchange of provably scarce, digital resources.

A more extensive discussion of digital currencies is presented, as an appendix, below.

The terminology surrounding monetary instruments on the blockchain can be confusing. For clarity, we will attempt to use terms systematically. The term *digital currency* will refer to any digital system which enables tracking of some representation of a monetary asset. The term *cryptocurrency* will refer to any digital currency which neither has intrinsic value, such as being accepted to pay taxes, nor external backing, such as being convertible for a set amount of gold. The term *crypto-fiat currency* will refer to any digital currency backed by some fiat (national) currency (or, by analogy, some commodity such as gold). The term *intrinsic coin* will be used for digital currencies, usually cryptocurrencies, used to pay fees in a blockchain system. The term *token* will be used for digital currency instruments managed by stored procedures.

Digital currency blockchain systems require some model for the representation of digital currency assets, some model to represent ownership, some mechanism for asset transfer, and some agreement over settlement. There are two current models for a currency management system[1]. In one model, the system can store each amount transferred (always in integer amounts of some sub-unit) as an asset belonging to some owner, the *unspent transaction output* (UTxO) model. In the other, *account* model, the system can give each owner an account for each given resource, record the amount held in the account, and then transfer amounts (again integer based) between owner accounts. The model of ownership differs in different blockchain systems, although usually, in open cryptocurrencies, ownership is assigned to public keys rather than to human identities, greatly simplifying the system. The transfer mechanism generally requires creating a digital signature for the transfer operation which validates against the owner public key. Blockchain systems mostly use irrevocable, immediate settlement in that all transfers are considered complete once included in the blockchain; however, realistic usage requires waiting for the system to extend the blockchain by a few blocks to ensure the transaction will stand. Even then, exceptional circumstances can lead to a collective modification of the blockchain history which invalidates the transfer. Settlement is conceptually instantaneous, realistically requires a period of confirmation, but, exceptionally, can be invalidated.

Digital currencies face a number of practical, economic, and legal issues, as discussed at greater length in the appendix. Cryptocurrencies face additional issues due to being monetary systems independent of the wider economy. Cryptocurrency monetary systems necessarily must have some defined *monetary policy* determining the amount of currency in circulation, the schedule under which new currency is generated (or existing currency withdrawn), the mechanisms by which circulation is controlled, and the policy of allocation of the new currency. Cryptocurrencies face additional legal issues since their lack of intrinsic value leaves open the interpretation that the coins are actually securities rather than a monetary instrument.

1 Joachim Zahentferner Chimeric Ledgers: Translating and Unifying UTxO-based and Account-based Cryptocurrencies 2018

Identity

The use of a blockchain asset tracking system to assign textual identifiers to owners arose as the first use of blockchains other than cryptocurrency. These systems assign ownership to on-chain textual assets much like cryptocurrency systems assign ownership of on-chain numeric values.

The blockchain project *Namecoin*[1] emerged in 2011[2] as an altered copy of the Bitcoin software focused on registering names and using a colored coin approach where the transfer of a particular pool of value represents transfer of the name. The project issues cryptocurrency tokens to miners and requires those tokens as fees to register and renew names. The project will assign a name to the first to request that name but requires the recipient to renew the name for a fee every 36000 blocks (around two hundered days). While originally mined as a separate project, the lack of participation of miners led to a change towards merged mining where Bitcoin miners could mine Namecoin blocks as a side-effect of their work on Bitcoin. Namecoin assigns domain names in a `.bit` top-level namespace and offers a Domain Name Service to resolve identifiers to JSON data structures with information about any registered identifier. Namecoin can also issue identity names. There exists a NameID service[3] built on top of namecoin which enables namecoin identities to be used as OpenID login credentials for web sites.

The blockchain project *Blockstack*[4] (originally called Onename) emerged in March 2014 as an alternative project for registering names. Blockstack was designed as an overlay system to be run on top of another blockchain project rather than being a standalone project in its own right. Blockstack originally was overlaid on the Namecoin project so that Blockstack transactions were special Namecoin transactions. Due to worries about the security of the Namecoin project, Blockstack decided to move to a new blockchain project and, in September 2015, moved to Bitcoin[5]. Incidentally, this move demonstrated the logical independence of the overlay system.

In 2018, the government of Dubai announced[6] that it would use a blockchain system to register businesses[7].

Registry: Fiat Currency

The use of a blockchain asset tracking system as a registry for deposits of external value, either fiat currency or precious metals, creates a digital asset with a known value which can be used as a currency. This has been explored by several groups, both private businesses and governmental institutions such as central banks. This application differs from pure cryptocurrencies; pure cryptocurrencies have no intrinsic value and do not require placing trust in any registrar.

The usage of crypto-fiat currencies in blockchain projects potentially allows a monetary instrument which is much less volatile than a pure cryptocurrency. Currency volatility limits usage of that currency, for example because it greatly increases the risk of taking out a loan in the volatile currency. However, the use of a crypto-fiat currency exposes the blockchain system to various international laws related to currency trading.

---

1 Namecoin
2 Wikipedia Namecoin Retrieved 2018-07-20
3 NameID
4 Blockstack
5 Muneeb Ali News is out *Twitter* 2015-09-12
6 Government of Dubai DED and DSOA unveil blockchain commercial registry project for improved ease of business Media Office 2018-05-01
7 Sujha Sundararajan Dubai Governement Unveils Blockchain Business Registry *Coindesk* 2018-05-02

This application of blockchain systems supposedly already exists, but trust in existing tethered currencies is undermined by lack of trust in the tethering process. For example, the *Tether* tokens on the Ethereum blockchain supposedly are matched to equivalent deposits of U.S. dollars, however the proof of registration is not convincing, leading many to consider Tether a scam. An influential paper makes the case that Tether is being used, without backing, to manipulate the price of Bitcoin[8]. Indeed, the United States government is currently investigating the owners of the Tether digital currency[9].

Central banks in various countries have been exploring this application of blockchain systems as a way to emit a national digital currency. Where private businesses would need to maintain deposits equivalent to the digital tokens issued, a central bank could emit these tokens only on its reputation, much like the emissions of fiat cash by central banks. However, these projects are still in the research phase due to the complexity of the consequences of such a move, the relative lack of experience with blockchain systems, and the conservative approach of central banks. Morten Bech and Rodney Garratt developed a useful taxonomy for the different kinds of financial instruments which could be issued by central banks[3]. The Bank of England has included the study of central bank digital currencies as part of its One Bank Research Agenda[4,5,6], leading to the *RSCoin* proof-of-concept[7] and to analyses of the macroeconomic impacts of such an instrument[8] (and the introduction of the concept of interest bearing digital currencies[9]). Similarly, Sweden's Sveriges Riksbank is considering issuing a blockchain currency as part of its eKrona project[10,11,12,13], although it is also considering issuing a digital currency in another form. The central bank of Canada, as part of project Jasper (CAD coin) issued a blockchain based coin for settlement between big banks[14]. Similarly, the central bank of Singapore, as part of project Ubin, developed a blockchain based, realtime gross settlement system[15]. While various central banks around the world continue to study issuing state backed cryptocurrencies, none have yet done so.

Registry: Real Estate

The use of blockchain asset tracking systems to define the ownership of real estate emerges as a natural extension of the ability to register and track assets. The United States' National Association of Realtors released a white paper discussing this approach in 2016[16].

Like any registry of real-world assets on the blockchain, a blockchain based real estate registry necessarily places trust in a third part. Some party necessarily must define the real estate asset to be registered and assign some on-chain representation of that asset. Furthermore, some party must

---

8 John Griffin and Amin Shams Is Bitcoin Really Untethered? 2018-06-13
9 Matthew Leising U.S. Regulators Subpoena Crypto Exchange Bitfinex, Tether *Bloomberg* 2018-01-30
3 Morten Bech and Rodney Garratt Central bank cryptocurrencies *BIS Quarterly Review* 2017-09-17
4 Bank of England One Bank Research Agenda 2015-02
5 Bank of England Digital currencies
6 Bank of England Primary questions
7 George Danezis and Sarah Meiklejohn Centrally Banked Cryptocurrencies 2015-12-18
8 John Barrdear and Michael Kumhof The macroeconomics of central bank issued digital currencies Bank of England, Staff Working Paper No. 605 2016-07
9 John Barrdear *personal communication* 2018-03-08
10 Cecilia Skingsley Should the Riksbank issue e-krona? Sveriges Riksbank 2016-11-16
11 Eva Julin *et al*. Riksbankens e-krona: 14 March 17 Project plan Sveriges Riksbank 2017-03-14
12 Eva Julin *et al*. The Riksbank's e-krona project: Report 1 Sveriges Riksbank 2017-09
13 Eva Julin *et al*. The Riksbank's e-krona project: Action plan for 2018 Sveriges Riksbank 2017-12
14 Rod Garratt CAD-coin versus Fedcoin R3 Reports 2018-04
15 Morten Bech and Rodney Garratt Central bank cryptocurrencies *BIS Quarterly Review* 2017-09-17
16 Mark Lesswing Blockchain in Real Estate: Useful Application Areas 2016-10

establish the identity of the person to which the asset representation can be assigned. The blockchain system can then track the changes in ownership by re-assigning the representation to another identity. The actual approach through which real estate and identity are assigned on-chain representations vary from project to project. Since these projects rely on trusted third parties for the actual registration, it is not clear to what extent these implementations require the distributed trust guarantees of traditional blockchains.

Many projects have been undertaken to use Blockchain systems for the registry and transfer of real estate in multiple jurisdictions around the world.

A proof-of-concept project in Honduras, undertaken in cooperation with Factom[1], was started in 2015 but eventually was abandoned[2].

Chicago's Cook county undertook a pilot program with Velox.RE[3] to test blockchain based ownership transfer of real estate[4]. The approach used by this pilot program used the colored coin approach, on the Bitcoin blockchain. This approach uses a specific token from the Bitcoin blockchain as a representation of the deed. Conveyance of that token then served as proof of the real estate transfer. Extra information about the deed, such as the address and name of owner, are recorded separately and a hash of that information is stored as part of the transaction transfering ownership of the token. In order to track these transactions, some code reviews the entire blockchain and extracts only those transactions which are part of the colored coins of interest to this registry. This approach was taken in the United States where a deed registration system is used for real estate transfers, unlike the title registry systems used elsewhere. This pilot program was apparently successful and therefore serves as an example for future projects.

The Swedish land registry authority, the Lantmäteriet, implemented a trial blockchain based real estate registry in partnership with ChromaWay AB[5] starting in 2016[6] which entered its second phase in 2017[78] and third phase in 2018[9]. This system used a private blockchain to store the record of transactions and used digtial signatures to secure the transfer. Since Sweden users a title registry system, the blockchain merely served as a common database for all participants in the process: buyers, sellers, the registry, real estate agents, and presumably lenders[10]. In middle 2018, the system was demonstrated to journalists[11]. This is apparently one of the most complete systems for real estate transfer using a blockchain system. Notably the system claims to be consistent with the new data privacy requirements of the General Data Protection Regulation of the European Union.

The National Agency of Public Registry of the Republic of Georgia, in cooperation with BitFury, also developed a trial blockchain based system for real estate registry and transfer starting in 2016[12].

1 Factom
2 Pete Rizzo Blockchain Land Title Project "Stalls" in Honduras *Coindesk* 2015-12-29.
3 Velox.RE
4 Ragnar Lifthrasir Permissionless REal Estate Title Transfers on the Bitcoin Blockchain in the USA!-Cook County Blockchain Pilot Program Report 2017-06-28
5 ChromaWay
6 Lantmäteriet The Land Registry in the Blockchain 2016-07
7 Lantmäteriet The Land Registry in the Blockchain - testbed 2017-03
8 Jonathan Keane Sweden Moves to Next Stage With Blockchain Land Registry *Coindesk* 2017-03-30
9 ChromaWay Blockchain and Future House Purchases: Thrid phase to be completed in April 2018 2018
10 John Camdir Swedent Conducts Trials of a Blockchain Smart Contracts Technology for Land Registry *Blockchain Magazine* 2016-06-23
11 Christine Kim Sweden's Land Registry Demos Live Transaction on a Blockchain *Coindesk* 2018-06-15
12 Stan Higgins Republic of Georgia to Develop Blockchain Land Registry *Coindesk* 2016-04-22

In 2017, this system was put into production[1] and by 2018 had processed over a million titles[2]. The government of Ukraine, also in cooperation with BitFury, attempted to use blockchain systems for real estate registry, starting in 2017[3]. Similarly, the government of Russia is experimenting with Blockchain based technology[4]. The state of Andhra Pradesh, in India, in cooperation with ChromaWay, is developing a pilot program for blockchain based real estate registry[5].

Multiple cities are testing the technology. The city of South Burlington, Vermont, in the United States in cooperation with Propy Inc. has also undertaken a pilot study on the use of Blockchains for real estate transfers[6]. The municipalities of Pelotas and Morro Redondo in Brazil, in combination with Ubitquity, are testing the use of blockchains for real estate[7].

The numerous projects undertaken so far testify to the interest in this approach. The success of various of these proofs-of-concept suggest that Blockchain projects are well suited to act as registries for real estate. While the current approaches are mostly experimental and each undertaken with its own approach, the common experience from these efforts could eventually lead to a standardized approach to the use of Blockchain projects for real estate registries.

Accounting

The use of blockchain asset tracking systems as the foundation of an accounting system is potentially one of the most interesting applications of the technology. There is great enthusiasm[8] and interest[9] in the use of blockchains for accounting. Accountants have even formed a coalition, the Accounting Blockchain Coalition[10], to push for this application of blockchain technology.

Accounting systems based on blockchains could replace the current accounting software. Such systems could, instead of maintaining the current state of the accounts, maintain the full history of transactions from which the current state could be calculated. In large organizations, all internal transactions could be validated by both parties taking part. External transactions would need to be validated as they are done currently, such as through receipts or other documents. However, such resources could be digitized and stored, while a hash could be included to immutably record the association of those documents with the transaction. The systems might even use a cryptocurrency like Bitcoin, although merely as a unit of account. Such systems could facilitate oversight and auditing by providing the full history of financial changes in the company.

Despite this promise, currently there are no accounting systems based on blockchains. Such systems will likely take time to emerge. Accounting practices change slowly. Keeping books is usually not the focus of operations so is usually not the focus of innovation in companies. The legal and regulatory issues involved in keeping books increases the cost of all changes. The entrenched nature of major accounting software providers makes it difficult for new approaches to take hold. Also, a major part of the benefits of blockchains to accounting would come from being able to cross-link

1 Laura Shin The First Governement To Secure Land Titles on the Bitcoin Blockchain Expands Project *Forbes* 2017-02-07
2 Shefali Anand A Pioneer in Real Estate Blockchain Emerges in Europe *The Wall Street Journal* 2018-03-06
3 Volodymyr Verbyany Ukraine Turns to Blockchain to Boost Land Ownership Transparency *Bloomberg* 2017-10-03
4 Nikhilesh De Russia's Government to Test Blockchain Land Registry System *CoinDesk* 2017-10-20
5 Nikhilesh De Indian State Partners with Blockchain Startup for Land Registry Pilot *CoinDesk* 2017-10-10
6 Peter Grant A Vermont City Tests Blockchain Technology for Property Deals *The Wall Street Journal* 2018-01-30
7 Garrett Keirns Blockchain Land Registry Tech Gets Test in Brazil *Coindesk* 2017-04-05
8 Jeff Drew How AI, blockchain, and automation will reinvent accounting *Journal of Accountancy* 2018-02-28
9 Deloitte Blockchain Technology: A game-changer in accounting? 2016-03
10 Accounting Blockchain Coalition

transactions between the accounting systems of separate counterparties, a benefit that can only arise after all the parties have their own blockchain based accounting systems.

Supply-chain management

The use of blockchain asset tracking systems to track the flow of goods through an entire supply chain have been repeatedly suggested as one promising area of application[1], though not without detractors[2].

In a supply chain, a distributed blockchain project which recorded the change of ownership of goods during transit could leverage the pairwise testimony of each exchange into a database that allowed everyone to establish the current state of ownership of (and possibly the location of) all the goods in the supply chain.

One of the most prominent application of blockchain asset tracking to supply chain management is a collaboration between the Maersk shipping firm and IBM[3], first as a collaboration[4,5], and then as a joint venture[6].

Stored Procedure Systems

The use of blockchain projects to store executable procedures offers the most powerful application of blockchain technology. In this application, the blockchain stores software code which can be invoked later by subsequent interactions with the blockchain project. The mechanism for storing procedural code on the blockchain grew out of the scripts provided as proof-of-ownership in Bitcoin transactions.

This usage revolutionizes blockchains, marking the second generation of blockchain systems. Instead of the blockchain system needing to integrate within its code explicit support for all the uses the system will offer, the blockchain system integrates a generic executable procedure system and that procedure system subsequently serves multiple new uses. This architecture effectively decouples the blockchain system itself from its uses. The Ethereum project was created explicitly with this design and was the first such system; Ethereum therefore serves as the poster child for stored procedure systems. One of the creators of Ethereum, Vitalik Buterin offered in 2014 an overview of different types of stored procedures and terms through which to refer to these types[7].

These stored procedures have been called *smart contracts*, a cute name for marketing. However, it is important to note that they are neither smart, since current procedures are all fixed function rather than adaptive, nor contracts, since they do not necessarily fulfill the legal requirements of a contract[8]. Defining and understanding the relationship between the formal computer code in a stored procedure and the legal requirements and responsibilities of participants remains to be fully

1 Bernard Marr How Blockchain Will Transform The Supply Chain and Logistics Industry *Forbes* 2018-03-23
2 Steve Banker Blockchain In The Supply Chain: Too Much Hype *Forbes* 2017-09-01
3 gCaptain IBM, Maersk Reveal Blockchain Solution for Global Supply Chain gCaptain.com 2017-03-06
4 IBM Maersk and IBM Unveil First Industry-Wide Cross-Border Supply Chain Solution on Blockchain IBM Press Release 2017-03-05
5 Tom Groenfeldt IBM And Maersk Apply Blockchain To Container Shipping *Forbes* 2017-03-05
6 Michael White Digitizing Global Trade with Maersk and IBM IBM Blogs 2018-01-16
7 Vitalik Buterin DAOs, DACs, DAs and More: An Incomplete Terminolgy Guide *Ethereum.org* 2014-05-06
8 Primavera De Filippi and Aaron Wright Blockchain & the Law: The Rule of Code YouTube 2018-04-27

understood. Primavera de Filippi and Aaron Wright recently published *Blockchain and the Law*[1] exploring many of these issues.

While stored procedures offer a remarkably powerful device for blockchain systems, the use of stored procedures is in its infancy so potential issues are poorly understood. Effective designs for procedures and the support required from the underlying blockchain system are still being elucidated[2]. Incorrect coding of the procedures can have disastrous consequences, as was seen when the collapse of *The DAO* (discussed below) lead to a split in the original Ethereum community. Even correctly coded procedures can have unintended impacts on the underlying blockchain system, as was seen with the CryptoKitties project (also discussed below) which overwhelmed the Ethereum blockchain. Finally, the legal, social, and ethical consequences of these systems have yet to be well understood.

Digital currency tokens

The use of blockchain stored procedure systems to issue and trade digital currencies has been one of the most successful applications of theses systems. Stored procedures can replicate the mechanisms for digital currencies provided in older blockchain projects without requiring any mechanism built-in to the blockchain itself. An entire digital currency mechanism can be built through a single stored procedure. The properties of these digital currencies are essentially no different than the properties of intrinsic coins hosted on their own blockchain projects. The only reason to maintain a built-in cryptocurrency mechanism in blockchain systems which support stored procedures is to link the use of the blockchain to fees paid for that usage. Apart from the change in mechanism, from built-in cryptocurrency code to stored procedure cryptocurrency code, there are no differences between the cryptocurrency token system described here and the cryptocurrency coin systems described earlier.

Ethereum, for example, currently hosts over a hundred thousand alternative cryptocurrencies[3]. This use has become so common that a standard has been developed, the ERC-20 standard[4,5], to formalize the behaviour of these cryptocurrency tokens and allow all such tokens to be traded on exchanges. There are numerous examples of stored procedure code for such tokens. The ease with which such tokens can be defined has led to an explosion of these tokens, many of which are released through initial coin offerings (ICO) to raise funds for particular projects.

Digital currency tokens can be issued as pure cryptocurrencies, that is with no inherent value, or as fiat-backed currencies, whose inherent value depends on the fiat currency backing the digital currency and which require some registration mechanism and party guaranteeing the fiat backing.

Non-fungible tokens

The use of a blockchain stored procedure system to issue and trade unique digital tokens which can not be mixed with other tokens follows the approach for digital currency tokens.

Ethereum, for example, started offering such tokens, and developed ERC-721 token specification to formalize the behaviour of such tokens. The use of these tokens to represent digital collectibles was

1 Primavera De Filippi and Aaron Wright Blockchain and the Law: The Rule of Code Harvard University Press 2018
2 Pablo Lamela Seijas *et al.* Scripting smart contracts for distributed ledger technology Cryptology ePrint Archive 2017-02-10
3 Etherscan Token Tracker Retrieved 2018-07-20
4 Etherum GitHub Repository ERC: Token standard #20 2015-11-19
5 Etherum Wiki ERC20 Token Standard

demonstrated conclusively by the CryptoKitties[1] project, which developed a generative system for images of kittens. The CryptoKitties contract was deployed as a stored procedure on the Ethereum network. Each unique token represents a unique visual image, differing mostly in facial expression and colouring. Additionally, two tokens can be associated and thereby generate a new, unique token. This allows for a system which generates new iterations. The procedure also creates a market for these tokens, allow the owners to sell them. The CryptoKitties project exploded in popularity creating a massive slow down in the underlying Ethereum network. In December of 2017, the traffic of this single procedure accounted for around a quarter of all the traffic on the network.

Escrow

The use of blockchain stored procedure systems for escrow has been repeatedly suggested.

For example, an escrow procedure could intermediate the exchange of resources between parties. The procedure would require that both parties both first transfer the resources to be exchanged to the procedure itself, at which point the procedure would send the resources to the opposite parties. If either party failed to transfer their resource, the procedure would return the resource which had been transferred back to the original holder. The escrow procedure would be trusted since its entire code could be inspected by all. Such escrow procedures could be increasingly complex to consider more constraints. They could include temporal deadlines or trusted intermediary parties with the right to assert the conditions of the escrow had, or had not, been met, but unable to actually claim the exchanged resources.

Lottery

The use of blockchain stored procedure systems for a lottery is an obvious application.

For example, a stored procedure could accept funds from participants and then, at some given signal, randomly select a recipient to receive the accumulated funds. The specific design of the stored procedure would determine what kind of lottery would be run and what its reward structure could be.

Distributed Autonomous Organization

The use of blockchain stored procedure systems for the creation of self-contained organizations able to collect funds and then spend those funds on projects selected by participants in the organization was one of the most innovative applications suggested for these blockchain systems. We do not yet understand the full consequence of the existence of this type of organization.

The first attempt to build such an organization was undertaken by the Ethereum community in the middle of 2016. *The DAO*, as it came to be known, collected funds during May 2016, eventually amassing around 14% of all of the Ether tokens then in existence accounting for a value of around $150 million[2] at the time. Within a month, a security vulnerability in the code of the procedure implementing the organization was exploited and control was lost over around a third of the organization's holdings. This spectacular failure eventually led to a split in the Ethereum community, to a fork of the blockchain project into two, and to a clear lesson on the danger of placing financial trust in the code of a blockchain procedure. Indeed, this failure was of such scale

1 Crytokitties Collect and breed digital cats!
2 Wikipedia The DAO (organization) Retrieved 2018-07-20

as to change the world of blockchains forever: no longer could one innocently argue that the code of the procedure must always be considered the final arbiter of what should be permitted.

The *plantoid*[1], an artistic hack by Primavera di Filippi and others, serves as a demonstration of distributed autonomous organizations. The concept is to have a contract on the blockchain which can collect funds and, when sufficient resources have been accumulated, pay artists to create a new version of itself. In its initial form it simply collected funds on the Bitcoin blockchain and was controlled by an artist collective. However, the eventual intent is to build a stored procedure which can accumulate funds from donors and let stakeholders elect when and how to disburse those funds to instantiate a new iteration of itself, thereby reproducing in an analog of a life form.

Distributed Applications

The use of blockchain stored procedure systems to provide backend services for applications is a current area of particular interest. Such decentralized applications, often called *Dapps* (or even *Đapps* to be cute), are being built for many different purposes[2].

Currently, blockchain systems can not function as the complete backend system for decentralized applications. Instead, most projects use a hybrid approach where a regular server provides the backend for most of the application while the blockchain serves as the backend for only a few, key components.

Multi-Blockchain Systems

The simultaneous use of multiple blockchain systems offers a novel use for blockchain systems. Currently, the only well defined interaction between blockchain systems is anchoring, where one blockchain project anchors itself by declaring the hash of a recent block on another chain. There is active research for a mechanism which allows cryptographically secured exchange of assets between blockchain systems. This area of applications is currently in its infancy.

Anchoring Concept

The simultaneous use of multiple blockchain systems enables the use of the guarantees provided by one, larger, more dynamic blockchain system to testify to the immutability of another blockchain system, *anchoring* that other blockchain to the first.

For example, a blockchain used on its own as a personal, data archive provides no immutability guarantee since, given today's hardware, it is trivial to alter the blockchain archive and then recompute an entire chain of hashes to obtain a valid, although altered, blockchain. A scientist keeping her laboratory notebook as records on a personal blockchain cannot show that this record is original. However, the scientist can prove immutability of the personal blockchain simply by recording, periodically, such as at the end of each day, the hash of the last block of the personal blockchain onto another, public, high traffic blockchain outside of the scientist's control. The guarantee of immutability of the public blockchain testifies that the hashes it contains are original, and those hashes, in turn, testify to the originality of the scientist's personal blockchain and therefore of the laboratory archive. By anchoring a personal blockchain data structure into a public blockchain project, the scientist demonstrates that the personal blockchain structure correctly archives the scientist's work.

1 Okhaos I'm a PLANTOID
2 State of the Đapps

Similarly, a time stamping service can record on its internal blockchain the hashes which testify to the existence of digital resources. By periodically anchoring hashes of blocks on the internal chain to another blockchain, such as Bitcoin's, the time stamping service can demonstrate the non-mutability of its internal chain and transitively to the original existence of the digital resources.

Sidechain Concept

The simultaneous use of multiple stored procedure blockchain systems for the exchange of assets between the systems is actively being pursued through the *sidechain* concept. The sidechain mechanism potentially allows the migration of assets between blockchains while retaining all the cryptographic guarantees of the assets on their original blockchain. This differs from trading, on one blockchain, an asset registered to an asset from another blockchain, since the registration necessarily relies on some third party.

The sidechain application requires two or more blockchain systems, each with a stored procedure for sidechain applications. If correctly designed and implemented, the mechanism would allow resources from one blockchain simultaneously both to be secured away and to generate a cryptographic testimony of that storage. The cryptographic testimony could be passed to a stored procedure on another blockchain which would release a tethered representation of the original asset on the new blockchain. This tethered asset could be traded on the new blockchain system. A full sidechain system would also allow the tethered representation simultaneously to be retired on the second blockchain and to generate a cryptographic testimony of that retirement which then could be used on the first blockchain system to release an equivalent amount of the original funds.

The sidechain mechanism is of great interest as one of the possible mechanisms for scaling blockchain projects. A successful mechanism would allow for resources on any chain to be secured, distributed to any number of sidechains, traded on those side chains, and returned to the original chain if desired. This would enable a blockchain to scale simply by generating copies of itself.

No working solution for the sidechain mechanism has yet been found. The mechanism would require a sophisticated cryptographic mechanism which has yet to be invented.

Many other uses of blockchain projects are possible. The discovery of these uses and the implementation of viable solutions enabling these uses ensure the underlying vibrancy of the field of blockchain systems.

Digisoft Research

In compliment to our review of blockchain technology, we also undertook a series of research projects aiming to find an effective application for blockchains viable as a business in Uruguay today.

The initial interest of our applied research centered on leveraging blockchain technology to improve the electoral process. As discussed in the chapter on blockchain information systems above, the idea of end-to-end auditable voting systems[1] has received much attention at least since the work of David Chaum[2]. Several blockchain projects have arisen to attempt to improve the electoral process including the Agora.vote[3] and Democracy.earth[4] projects. However, the inherent tensions in the voting process make difficult its transition to the digital world. Voting involves a delicate interplay of transparency and secrecy to protect both the voter and the process: the rules of the voting system must be transparent and understandable to all, the eligibility of voters must be transparent and widely accepted, the actual vote must be secret and it must not be possible for anyone, the voter included, to show how the vote was cast, and the counting of the vote must be transparent, repeatable, and auditable. The problems with automated voting machines have been widely discussed and led to the publication of guidelines for state actors[5]. The complexity of cryptography for the general citizen makes difficult its use in a system which must be understandable by all. None of the existing blockchain projects have been used in standard elections. Overall, the issues involved in applying blockchain technology to voting, such as applied cryptography, voter education, and electoral design, fall well outside of the expertise of Digisoft; we therefore felt it would not be productive for us to pursue this line of research.

A second interest of our applied research considered applying blockchain technology to developing predictive markets, hopefully finding some way to include the recent ePeso experiment[6] in the design. Predictive markets expose the expectations of the crowd in real-time which itself offers value. Several projects, such as Augur[7], have been announced using blockchain systems in predictive markets. However, the lack of any clear example led us away from this application area. The core complexity of predictive markets comes from their being essentially financial markets; our lack of experience in this area argued against adopting this focus. We did not have any clarity of what types of predictions our market should be trading. None of the blockchain systems working in this area have working, finalized implementations to study so that all this work would have to be *de novo*. The ePeso system, since abandoned, replicated existing bank accounts without offering any of the benefits of blockchain systems and thereby had limited interest for us in this context. We therefore decided that we would not be able to complete any productive research in this area within the time frame of the funded research effort.

A third interest of our applied research explored the design of a blockchain project providing a cryptocurrency pegged to the value of the Uruguayan Peso within the system of payments in Uruguay. The broad utility of such a project seems apparent, so much so that we consider that such a system ought to be run by the government. However, we focused on the private development of

1 Wikipedia End-to-end Auditable Voting Systems Retrieved 2018-06-15
2 David Chaum Secret-Ballot Receipts: True Voter-Verifiable Elections IEEE Security and Privacy 2(1):38-47, 2004
3 Agora.vote
4 Democracy.earth
5 US Election Assistance Commission Voluntary Voting Systems Guidelines US EAC, 2009
6 Plan piloto ePeso
7 Augur

such a payment system. Our research effort exposed some of the issues involved in running such a system in Uruguay. The legal context around such a system includes conflicting goals of financial transparency and secrecy of individual activity which would need to be resolved. The economic structure of such a system could take many forms including being based on collection of fees, on extracting gains from market spreads, or even being absorbed entirely as a subsidy or a loss leader. The technical design of such as system is relatively simple, replicating in large part the design of the Bitcoin or Ethereum systems. This interest is developed as one of the sections below. As part of this research effort, we initially imagined developing a prototype implementation but have since abandoned that idea: we could find no design which demonstrated a useful aspect of such a system, was implementable in the time frame of this research project, and was not completely trivial.

A fourth interest of our applied research explored a vision of a future Uruguay in which many, independent blockchain projects exist. In such a Uruguay, blockchain projects with formally recognized legal standing would provide an infrastructure useful to unrelated, independent blockchain projects. This research therefore examined the utility, applicability, and design of a national blockchain infrastructure for Uruguay. A national blockchain infrastructure could underlie archival systems using the integrity guarantees of cryptographically self-referential data structures. A national blockchain infrastructure could leverage the transparency provided from the open sharing of transaction ledgers to improve oversight and involvement. A national blockchain infrastructure which included a national digital currency could form the financial bedrock for the government, for businesses, and for individuals. This interest is developed, in the abstract, as one of the sections below.

A fifth interest of our applied research attempted to develop a software code base for generating and running prototype blockchain systems sufficiently rigorous yet simple enough to serve pedagogically, sufficiently flexible to generate systems with different purposes and functions, and explicitly able to explore interactions between blockchain systems. Existing software code bases, even the most flexible, facilitate the building of single blockchain systems but not the building of multiple, interacting blockchain systems. Our own software code base, the Meristem project, serves for experimentation in blockchain projects and their interaction. We focus on information flow and therefore seek minimalist designs so as to discover the rules, data structures, and message exchanges which are strictly necessary for each piece of functionality. Our software code explicitly ignores the detailed work required to secure such a system cryptographically in order to avoid that complexity; our code therefore can not serve as the basis for any actual, functioning blockchain system. This work is at an early stage, however, even now, the project has revealed an innovative direction for blockchain systems of the future: blockchains which explicitly define, within themselves, the rules under which they operate. The Meristem project is presented as the last section below.

Blockchain for Payments

This research effort examined the design of a new payment system in Uruguay based on a cryptocurrency token whose value was pegged to the Uruguayan Peso and which was exchanged through a blockchain based network.

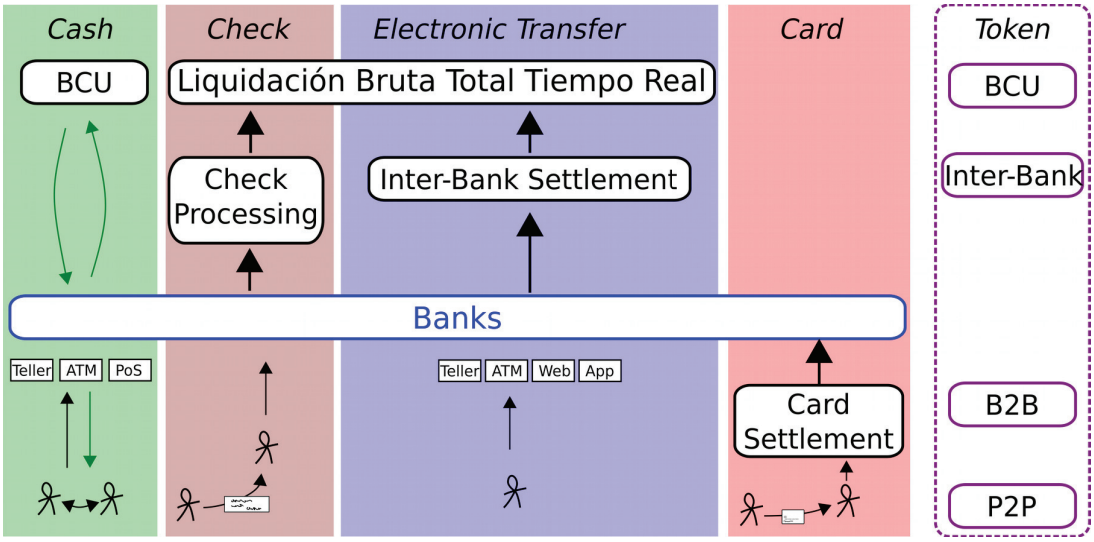
Our starting concept was of an independent company operating as a bank or similar financial institution and providing a payment platform enabling the direct exchange between two persons of a cryptocurrency token. We imagined ensuring that token had a stable value by pegging the token to

the Uruguayan Peso through a full deposit mechanism, possibly in an account at the central bank, and by providing a permanent guarantee of convertibility between the token and fiat money. The core functionality of the system to end users could be as simple as that of Bitcoin.

We originally planned to develop a full design for this system and use that design to build a simple prototype. However, as explained below, we were not able to establish sufficiently the design constraints to complete a design nor were we able to design any prototype worth developing. Our research effort therefore ended with this presentation of this partial design.

Uruguay's system of payments

The existing system of payments in Uruguay uses several monetary instruments: cash (banknotes and coins), checks, electronic transfers, and cards or electronic money. For each of these, a different chain of settlement systems come into play, chained all the way up to the central bank, the *Banco Central del Uruguay* (BCU).



This diagram is derived from the latest report on the payment system in Uruguay published by the central bank[1]. The system of payments using cash involves the minting of these instruments by (or on behalf of) the central bank, the sale and transfer of these instruments to banks, the sale and transfer of these elements to individuals through tellers, automated teller machines (ATM), or as 'cash back' at business points of sale (PoS), and then the use of these instruments for payments. The system of settlement using cash involves the physical transfer of the monetary instrument itself. Notably, cash is the only instrument in the current system which allows direct settlement between end parties without passing through the banking system. The system of payments for checks involves the transfer of a paper check between end parties, the deposit of that instrument (or, these days, its image) at a bank, the transfer of that instrument (or image) to the national check processing system with eventual inter-bank settlement via the central bank's *Liquidación Total Bruta en Tiempo Real* (LBTTR) settlement system. The system of payments for electronic transfer involves end user manipulation of their accounts with possible settlement between banks through their inter-bank settlement system or through the central bank's LBTTR. The system of payments for cards, debit or credit, first uses the Urutec card settlement system and then the regular banking system

1 Banco Central del Uruguay, *Reporte Informativo Nº17*, *Sistema de Pagos Minorista*, §3(p5), 2017

with electronic payment settlement thereafter. This system is also used by current providers of electronic money who mostly structure the design of their systems on the card payment system.

Blockchain technology conceptually allows the introduction of a new, independent payment system based on a cryptographic token as its monetary instrument. Such a system could operate at several levels in the settlement layer: as a final settlement at the central bank between itself and the accounts of major actors, as an inter-bank settlement layer, as an inter-business settlement layer, or as a payment layer between end parties. The blockchain based token exchange system would improve the system of payments in Uruguay in several ways. This new system would add robustness to the overall system of payments by being independent of the banking system. The new system would provide the ability for direct settlement between end parties like when using cash but with all the modern convenience of a digital electronic transfer system. The new system might also provide for improved transparency since the system inherently requires validating the entire transaction history contained in the blockchain, potentially improving the oversight of the central bank or transparency for citizens at large. These significant benefits make it worthwhile to explore such a blockchain based payment system for the future of Uruguay.

Uruguay's legal framework

The easiest approach to implementing a cryptographic token based payment system, if it were not done by the central bank itself, would be through some independent institution. Any institution providing a blockchain based, token exchange system for Uruguay would need to comply with the framework of laws, rules, and regulations established by Uruguay itself or applicable to the country from abroad. The legal context therefore provides a fundamental source of constraints on the design of the blockchain system by establishing the requirements of the institution, of its interaction with end users, of its financial instrument, and of the information flows necessary to comply with the laws. A complete review of the legal framework constraining an institution providing a token payment system lies outside our domain of expertise and requires a work effort much larger than available in this research effort: here we undertake only a cursory review.

Uruguay, starting in 2014, modernized its legal system explicitly to allow for 'electronic money' (*dinero electrónico*) which could be used to innovate and modernize the system of payments in the country. These changes also introduced a new class of financial institution, beyond banks, allowed to issue electronic money and provide a payment system using that instrument which are labelled *Institución Emisora de Dinero Electrónico* (IEDE). Therefore, a natural design for an institution providing a blockchain based, token exchange system would be as a registered IEDE.

The legal framework surrounding IEDE entities is difficult to reconcile with the design of a blockchain system. IEDEs were allowed in 2014 by law 19210[1]. These institutions must follow both presidential decree 263/015[2] of 2015 and the regulations passed by the central bank, notably those collected in book VII[3] of the collected regulations of the central bank. The law itself and the regulations decreed by the presidency do not appear to conflict with the creation of an IEDE providing a cryptocurrency, especially if it did not handle salaries, benefits, retirement, or disability payments directly, since handling these brings on many extra requirements. However, the law,

1 Parlamento del Uruguay, Ley Nº 19210 *Ley de Inclusion Financiera*, Uruguay, 2014
2 Presidencia de la República, Decreto Nº 263/015 *Reglamentación de la ley 19.210 (Ley de inclusion financiera) relativa al acceso y uso de servicios financieros y la transformación del sistema de pagos*, 2015
3 Banco Central del Uruguay, Recompliación de Normas, Sistema de Pagos, Libro VII, *Medios de Pago Electrónicos e Instituciones Participantes del Sistema de Pagos*, 2018-01-03

presidential decree, and especially the regulations of the central bank are designed implicitly with a structural model which does not align with the decoupled nature of a generic blockchain based system (as presented above). For example, the very name of the IEDE institution within the law, suggests that a single entity both provides the payment platform to end users and guarantees the backing funds for the electronic instruments, when these are not necessarily joined. The central bank regulations in particular rely on some pre-existing, but undefined, model of the payment system. Law 19210 discusses only four types of institutions: financial intermediaries (i.e. banks) accepting only general funds, IEDEs accepting only general funds, banks accepting both general funds and also salaries or other regular benefit payments, and IEDEs accepting both these types of funds. The presidential decree maintains this focus on four institutional types, focusing primarily on the latter two. However, the regulations of the central bank define, in Article 80, a slew of new institutional entities (*emisores*, *adquirente*, *sello*, *processador*, *adminstrador de PoS*, *switch*, and *proveedor)* whose relations to the former four entities are not entirely transparent and whose relations to the actors in a blockchain based payment system remains undefined.

The legal framework also includes many other laws which could potentially impact an IEDE providing a cryptocurrency. For example, the international financial control rules, such as the Know Your Customer (KYC) rules and Anti-Money Laundering (AML) rules would place a significant burden on any IEDE dealing directly with end users. Similarly, laws protecting personal data such as Uruguay's law 18331[1] and the European Union's General Data Protection Regulation[2] will necessarily apply, apparently in direct conflict with the rules for financial control.

Legally, then, much work remains to clarify the possibility of forming an IEDE which provides a blockchain based payment system. Undoubtedly, this work would require negotiations with the central bank and might even require rewriting parts of the legal framework for these institutions. All such work lies beyond our purview and outside of our domain of expertise.

Our failure to determine the full impact of the law on the design of the institution, on the interactions with end users, and on the flow of user information severely cripples our ability to design an actual information system for the exchange of cryptographic tokens on the blockchain.

An incomplete design

The design of a system of payments based on a cryptographic token using a blockchain system would require the design of the institution implementing the system, of the monetary instrument exchanged through the system, of the blockchain system itself, including its rules, actors, network, and code, and of the interaction of this system with the outside legal and economic context.

The institution implementing this blockchain based payment system could be a part of the national government or be some independent entity.

Many reasons suggest that this payment system should be implemented as part of the national government by the central bank (BCU) itself. As explained below, the token would necessarily be pegged to the Uruguayan Peso and would therefore essentially be a financial instrument of the

1 Parlamento del Uruguay, Ley No 18331, Ley de protección de datos personales, 2008-08-18

2 European Parliament and Council, Regulation (EU) 2016/679, *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*, 2016

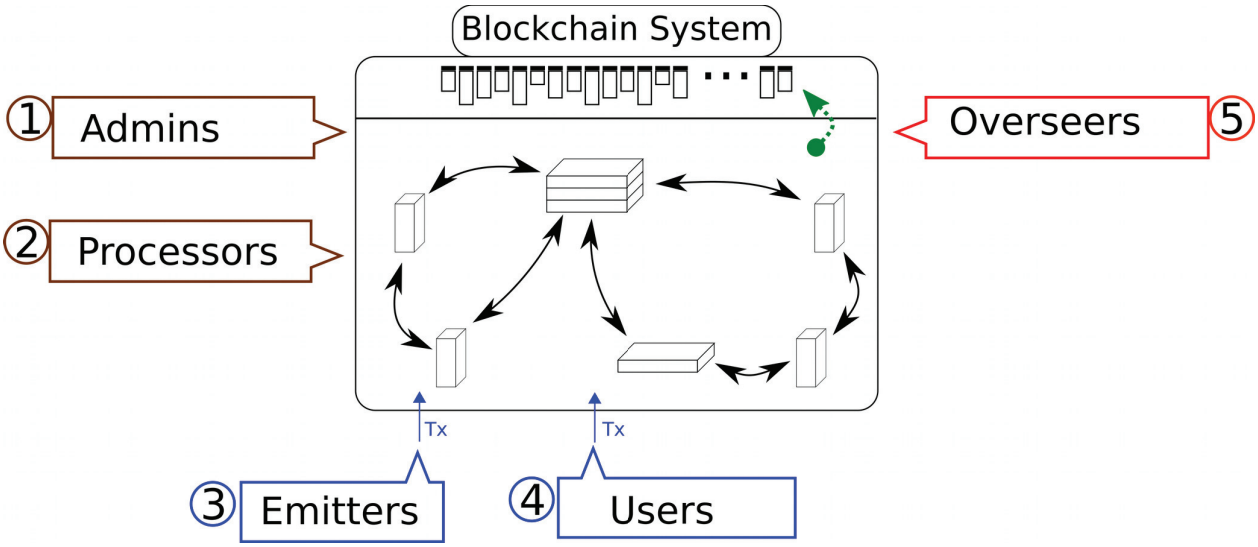
central bank. Since the value of the token rests on trust in the central bank, the blockchain system could leverage this trust relation to greatly simplify the overall system. These various advantages led us to dedicate substantial research into a central bank based design which we present below.

Alternatively, the payment system could be implemented by an independent institution. This would need to be a regulated financial entity, either a bank or an IEDE. Since the former include substantial activities and requirements unrelated to this research, we consider a design where the institution implementing the system operates as an IEDE registered with the central bank. An effective design for such an IEDE would necessarily include establishing its business plan showing the economic structure which would allow it to survive. Economically such an institution could maintain itself directly through fees, through spreads, or through other funds, for example, if it were subsidized as a loss leader providing a service of value to end users. Here, we skip all considerations of the economic structure of the institution, assuming simply that some viable economic model could be found.

The monetary instrument used in a blockchain based payment system necessarily would be a cryptographic token. Due to the legal requirements that IEDEs not undertake fractional reserve operations and provide full backing to any electronic instrument they issue, the cryptographic tokens would need to be fully backed by fiat funds held on reserve in some bank account. The easiest way to conceptualize this would be to have a token whose value is pegged directly to the Urguayan Peso. The system would require that any emission of cryptographic tokens be directly tied to a deposit of fiat funds and, presumably, that this relationship be regularly and transparently audited. Note, that the actual emission of tokens and the backing by deposits of fiat currency need not be undertaken by the same institution implementing the blockchain system. Unlike the *a priori* conceptualization of the law, the entity emitting fully-backed electronic money can be completely independent from the entity providing the system of exchange for that money.

Realistically, any such blockchain based payment system in Uruguay would want to offer cryptographic tokens for each of the monetary units of accounts used in the country: the Uruguayan Peso, the United States' Dollar, the Uruguayan *Unidad Reajustable* (UR) currently worth around a thousand pesos, and the *Unidad Indexada* (UI) currently valued around four pesos. However, since the design of the system would be similar for each of these units, we can focus on only one as a starting point and we choose the Peso.

A blockchain system for payments using a cryptographic token pegged to the Uruguayan Peso would involve all of the elements of any blockchain network. The core functionality could be simple and similar to the Bitcoin system, although substituting a new mechanism for the emission of tokens instead of Bitcoin's 'coinbase' transactions, eliminating transaction fees, and avoiding Bitcoin's inefficient proof-of-work consensus mechanism. The details of the system would need to be defined including the rules for the self-governance of the system, the rules for the actors in the system, the technical rules for operating the network, for processing transactions into new blocks, and for validating the blockchain, and the rules for operating the system within the legal jurisdiction and business environment of Uruguay today.



A blockchain system maintains the blockchain as a shared database, accepts submitted transaction requests, and periodically emits a new block by validating the blockchain, calculating initial state, selecting a pool of transaction requests, validating the transaction requests in order, and generating a new block with the valid transactions. The blockchain system relies on the rules governing the system, the code running on the nodes, the processing nodes running the code, and the shared blockchain database. If the system required authentication or identity based authorization, an access control sub-system would have to be included.

The roles of actors in the blockchain system, generically, fall into three groups: providers of the system (admin, processor), users of the system (both emitters of tokens, and owners and traders of tokens), and overseers of the system, either internal, governmental, or from the general public. Here, these roles are discussed separately based on their independent impacts; nothing prevents one actor from performing several roles in the system. Indeed, apart from end users who would always be multiple, the system could be provided by a single actor who acted as administrator, processor, emitter, and overseer.

The administrators of the blockchain system establish the rules under which it operates, provide the software code (or implementation standards) for the network nodes, and oversee the network. In our conceptualization, this role would fall to the institution registered with the central bank. This institution would need to combine system operations to maintain the network with development operations to develop, maintain, and extend the software used in the network. The rules of operations of this administrator would have to be clear and transparent to everyone.

The participants in the network provide machines to act as nodes which maintain the network, process transactions, emit new blocks for the blockchain, and share the blockchain database. The network could potentially be open to many participants since all are cross-validating each other. At its most trivial, this network could consist of a single machine. However, for robustness, the network should be distributed between multiple participants and spread over the entire national territory.

The emitters of cryptographic tokens would couple the system's mechanism for defining new tokens with the deposit requirements accompanying the emission. At its most trivial, this could be a single

institution, emitting a single pool of tokens possibly as part of the genesis block, backed by a single deposit, and establishing the entire monetary supply for the payment system. More realistically, the system could allow such emission events to be undertaken as needed to control the money supply, possibly by multiple separate institutions. The system could also allow for the withdrawal of tokens and recuperation of equivalent amount of fiat currency from the secured deposits. Regardless, the system would require regular auditing ensuring that the amount of funds circulating as tokens in the system match the amount of funds on deposit.

Many types of users of the payment system are possible, ranging from direct participants in the network (who maintain their own processing nodes) to users indirectly participating through intermediary institutions. Such intermediary institutions could operate in several ways, either holding actual tokens in user accounts, providing internal accounts backed by reserves of tokens, or even providing partial ownership interests in an overall pool of tokens. Like all intermediation, there is no limit to the potential complexity of these relations. Such intermediary institutions provide one avenue for resolution of the legal conflicts between needing to know the customer, the requirement for blockchain transparency, and the requirement for protecting private data and deleting it on demand-the institution could hold and manage end user data while the network could trade between anonymous accounts.

The oversight of the payment system can be performed simply through the review of the blockchain data structure. This oversight could be purely internal, could be governmental, for example by the central bank, or could be open publicly allowing for complete transparency of the system.

An actual implementation of this sort of blockchain payment system would also need to allow the interchange of payments between this system and the existing payment systems. This presentation does not consider this interchange.

All of these actors, in any viable system, would necessarily need to have the correct financial incentives to ensure their ongoing participation. However, we do not consider these financial constraints here.

The software implementation of this blockchain system could be built using several approaches. A new code base could be developed specifically. If not for the fees involved, the implementation could actually be done using one of the existing cryptocurrency projects, for example, by creating an ECR-20 token on the Ethereum blockchain. Alternatively, the software code of an existing project such as Bitcoin or Cardano could be adapted. The easiest way to would probably be to use a Hyperledger Fabric implementation, possibly built using Hyperledger Composer.

The futility of any prototype

We originally intended, as part of our research effort, to leverage our design of a token based payment system to develop a prototype implementation. We started this work by developing a set of use cases which would reflect the basic functionality of the system, would demonstrate the integrity of system results, and would show the contributions of the different actors. However, after extensive examination, we could not develop any design for a prototype whose implementation would provide value either to us or to others.

A core impediment to our design and implementation of a prototype came from our inability to define the actual constraints operating on our design. For example, the contradictory requirements of banking rules requiring transparency to avoid criminal exchanges of currency and of customer protection rules requiring privacy make it unclear what information must be tracked and if maintaining any permanent, public database of transactions can be made compatible with the requirement to privacy since the blockchain history, even one using anonymous public keys as identifiers, can be de-anonymized through analysis. Without fully defined constraints, the diversity of allowable implementations makes arbitrary the selection of a specific alternative and makes difficult the creation of concrete rules in software code.

A second impediment came from the triviality of our use cases. The simplest use case we could imagine would be the transfer of funds held by one user to another but this use case, on its own, provides almost no value. At the level of the information system, a transfer of funds merely requires validating that there are sufficient funds to transfer and then updating two balances. This can be achieved trivially so its demonstration provides no value; indeed, this use case is demonstrated in every Bitcoin transaction. However, expanding this use case into something worthwhile, for example showing that the full integrity of the payment system is maintained after the transfer, would require a massive undertaking including establishing a full threat model for the system and demonstrating the cryptographic security of the system. These are full research efforts in their own right.

A third impediment of implementing a prototype is that we are conceiving of a single institution implementing the payment system but proposing the use of a blockchain based design whose true value only emerges when applied to a system run by a distributed set of independent, mutually distrusting actors. This suggests the design should really be for a network built by a cooperating set of businesses rather than for a single institution.

These impediments led away us from implementing a prototype. Without a full set of design constraints any prototypical design would have little relevance to any real world scenario. With only trivial use cases, any demonstration of functionality would be silly. Indeed, such demonstrations are generally undertaken as the first example implementations for developers exploring code providing blockchain implementations such as the examples of the Hyperledger Composer project[1]. Without a design in which many independent actors are involved (but thereby in conflict with the idea of an IEDE), a blockchain based design makes little sense.

Instead of developing a trivial prototype of little actual interest, we decided to extend our analysis on the legal, economic, business, and technological context for the implementation of a blockchain based payment system. This led us to focus our research on the analysis just presented and to focus our programming work on the Mersitem project presented below.

1 Hyperledger Composer Playground Tutorial Retrieved 2018-07-10

A Uruguay of Blockchains

Blockchains, for better or worse, have already arrived in Uruguay. Bitcoin, Ethereum, and other systems, since they are open to all Internet users, can be used from Uruguay. Private blockchain projects are being designed. The government, businesses. and individuals of Uruguay all now face the decision of how to react to the existence of blockchain technology: ignore it, fight it, adapt to it, or adopt it.

Adopting blockchains could benefit many parties in Uruguay: governments, businesses, and individuals.

Government could benefit from using blockchain systems. The legislative branch could maintain official records of their sessions and formally register the changes to the law as transactions on, or validated by, an official blockchain. The executive branch could document appointments, record changes in regulation, and document financial flows through ministries and to contractors. The judicial branch could register their dockets and decisions. This would complement the current system of recognizing government events only when published in the official record by the *Dirección Nacional de Impresiones y Publicaciones Oficiales* (IMPO)[1]. Furthermore, the entire infrastructure of official records, including the real estate registry, automobile registry, and civil status registry could evolve to use blockchain technology to ensure archival authenticity of all digital changes. The government could also introduce a digital currency, to serve as a store of value equal to the Uruguayan Peso, as a medium of exchange contributing to the national system of payments, and as a unit of account for government ministries, government owned enterprises, or even non-governmental entities. Perhaps even the voting process could be archived on a blockchain.

Businesses could benefit, individually and collectively, from using blockchain systems. Individual businesses could record their official decisions in an unalterable ledger. Notaries could keep digital books in parallel with their official paper records with the same integrity guarantees. Businesses could record their internal finances as cryptographically secure changes to a central ledger ensuring transparent auditing. Business contracts could become documents secured through cryptographic hashes or even, in some simple cases, be implemented as software instructions within blockchain systems. Consortia could form to track events throughout a specific market, such as the livestock markets, or through the entire supply chain of an end product, such as a food or medicine.

Individuals could also benefit from using blockchain systems. Independent professionals could use blockchain systems to record and timestamp their official business documents, such as bids, bills, and receipts. Inventors, authors, artists, and musicians could document the existence of their creative works through affirmations made to blockchain system. All users could benefit from a digital system for financial transactions.

We envision the future of Uruguay as including multiple blockchain projects: the global, open cryptocurrency systems such as Bitcoin, Ethereum, and Cardano, private, permissioned blockchain systems currently being developed in industry, and special purpose blockchains servicing the needs of collaborations or of individual entities.

1 Dirección Nacional de Impresiones y Publicaciones Oficiales

National Infrastructure

We propose that Uruguay run an official blockchain system as a national infrastructure for general use and as the foundation for the innovation of other, custom blockchain systems. We further propose that the government undertake various legal reforms to give judicial authority to these systems.

One proposal is for a blockchain system open to all for the recording of small amounts of user-chosen data, a 'declaratory' blockchain system. This declaratory blockchain system provides a way for government, business, or private actors to archive data officially. The data can be recorded either directly, as the data being recorded on the declaratory chain, or indirectly, by recording the cryptographic hash of some digital data on the 'declaratory blockchain' and archiving the digital data itself separately. In both cases, the declaratory blockchain serves as a proof-of-existence at a given time of the digital data. A declaratory blockchain such as this provides fertile ground for innovation in the storage, review, and auditing of archival records.

Another proposal is for a blockchain system for the exchange of digital tokens endowed with a fixed financial relation to some currency recognized by the state. The currency against which to peg the digital token is arbitrary and a system could be designed to handle the Uruguayan Peso (Peso), the Unidad Indexada (UI), or even the United States Dollar, either individually or in combination. For simplicity in our discussion here, we consider only a token whose value matches the Peso. A system for the ownership and exchange of such a cryptoPeso digital token offers many routes to innovation in the financial market through all three functions of a currency: as store of value, as medium of exchange, or as unit of account.

These two proposed blockchain systems, while independent, offer a base blockchain infrastructure on which third party projects could be developed.

Several reasons lead towards a design in which these two blockchain systems actually are combined into a single system. First the two functions are completely independent and therefore do not conflict. Second, the integrity guarantees of an archival blockchain system strengthen as the amount of separate data recorded increases; the declaratory blockchain would therefore benefit from the inclusion of financial transactions in the same blockchain. Third, a digital token system provides a natural approach for payments for the declaratory blockchain. For example, the providers of the declaratory blockchain system could require payment of one cryptoPeso in order to submit a declaration transaction, expensive enough to eliminate frivolous use (spam), cheap enough to not present any hurdle to adoption, and transparent enough to be easily understood by all.

Several reasons suggest this national infrastructure should be run by the national government although the infrastructure could be maintained by a consortium. A declaratory blockchain system is relatively simple to create and maintain, easily within the technical and financial reach of private enterprise. However, since the entire purpose of these archival declarations would be to have irrefutable legal standing, it would make more sense for such a system to be run by the state. Many agencies could be responsible. To the extent that the declaration system acts as the official record, it could be taken on by the official government printing office, the *Dirección Nacional de Impresiones y Publicaciones Oficiales* (IMPO)[1]. Alternatively, it might be run by the *Agencia de*

1 Dirección Nacional de Impresiones y Publicaciones Oficiales

Gobierno Electrónico y Sociedad de la Información y del Conocimiento (AGESIC)[1]. A financial blockchain system tracking digital tokens pegged to a fiat currency could also be developed by private enterprise: the system could be as simple as Bitcoin and could be backed financially by a pool of fiat currency stored in a bank and regularly audited. However, this system acts essentially as the digital equivalent of cash and therefore falls naturally to the role of the central bank. Furthermore, the central bank, the *Banco Central del Uruguay* (BCU)[2], recently showed its interest in exploring a digital alternative to cash through its ePeso experiment. Since many central banks around the world are exploring the use of blockchain technology as a system through which to offer digital cash, it seems natural for the BCU to undertake this role.

Since both the declaratory blockchain function and the digital currency blockchain function could be combined into a single blockchain system, and since the digital currency function naturally belongs under the control of the central bank, it seems most natural for the BCU to run a single blockchain system providing both functions. The legislature should adapt the laws of Uruguay to permit this and to give legal weight to these blockchain interactions.

Declaratory Blockchain

A blockchain system on which users can record arbitrary data provides value beyond, or outside of, the inherent purpose of the blockchain *per se* and provides a foundation which can be leveraged by other blockchains. Such a blockchain system, especially one which had official government recognition with legal standing, could provide a useful foundation for a whole ecosystem of activities and other blockchains.

One important contribution of a blockchain system which allows the recording of arbitrary data is to provide an archival proof-of-existence and timestamp for that arbitrary data. The replication of the blockchain containing the data between all the nodes in the blockchain system ensures the preservation of the archival proof. The presence of the data between various timestamps in the blockchain establishes a time after which and another before which the data can be recognized to exist. Usually, in order not to bloat the blockchain, one records a compact, cryptographic hash of the digital resource to serve as a proof-of-existence for the resource rather than recording the resource itself.

Such timestamping services already exist, provided, for example, by Surety[3] or GlobalSign[4]. Surety provides a timestamp service by receiving a cryptographic hash of some digital resource, signing that hash with a certificate returned to the user, chaining the hash into Surety's database, and periodically publishing a hash of the state of the Surety database into a newspaper. Thus the cryptographic hash attests to the existence of the resource, Surety independently attests to the existence of the hash, and the newspaper publication attests to the state of the Surety database on the day of publication preventing retroactive modification. More recently, the Bitcoin blockchain has been used in a similar way. In that case, a user merely needs to record the cryptographic hash of the digital resource through some Bitcoin transaction. Once the hash is recorded on the official blockchain, the digital resource can be considered to be timestamped by the value of the time elements in the header of the next few blocks. (The lack of perfect synchrony between machines

1 Agencia para el Desarrollo del Gobierno de Gestión Electrónica y la Sociedad de la Información y del Conocimiento
2 Banco Central del Uruguay
3 Surety
4 GlobalSign

makes the timestamp only approximate.) Either of these systems can be reliable for time stamping digital resources. The former is more complex and involves a third party. The latter only involves an external blockchain system. Both, however, suffer from being unofficial. In any legal dispute, either system would be required to justify their internal workings.

An officially recognized blockchain that allowed for the recording of arbitrary data could serve as a formally recognized time stamping service. Official recognition of the blockchain would eliminate questions as to its operation. Formal recognition of its authority for time stamping would ensure the legal standing of the timestamp as testimony to the existence of a digital resource in any court proceeding. This service would provide an intermediate function between the official journal and the general public announcements made in press conferences or through social media platforms such as twitter. We can imagine the goverment, its ministries, governmental companies, and even private entities adopting the timestamping service to record official milestones: appointments to office, calls for tender, registration of the reception of bids, declarations of awarded contracts, or changes to public policy.

Another, similar, contribution of a blockchain system which allows recording of arbitrary data is to anchor a separate blockchain project. For example, a blockchain system used internally in a business or run by a consortium could periodically publish some cryptographic hash attesting to the state of the blockchain on another, more public blockchain. For example, a business could run its accounts using some internal blockchain system then, by writing the value of the hash of the latest block in the internal chain to some public chain, the business can demonstrate that it has not altered its past accounts. This approach leverages declarations on an external chain to act as public witness to an internally controlled chain.

CryptoPeso Blockchain

A blockchain system using which users could possess and exchange digital currency tokens backed by Uruguayan Peso fiat could provide both a useful tool for payments and a platform for future innovation. Even more useful would be if the system also allowed the exchange of other fiat monies in common use in Uruguay such as the *Unidad Indexada* (*UI*)[1], *Unidad Reajustable* (*UR*)[2], and US Dollars.

The basic design of a national blockchain for payments could follow the design outlined above. Ideally, the system could isolate the blockchain payment system from the crypto-fiat currency issuance and withdrawal mechanism.

The existance of such a system would provide a platform for innovation. Having a fiat backed, government approved digital currency would bring with it the three core functions of money: as store of reserves, as medium of exchange, and as unit of accounting. The latter could prove broadly attractive since accounts could be kept secure simply by calculating them directly on a blockchain.

The design of such a system would require extensive investigation by the government. Offering a payment system potentially raises many legal issues of international scope. Payment systems today must integrate limitations due to Know Your Customer (KYC) and Anti-Money Laundering (AML)

1 Instituto Nacional de Estadística Unidad Indexada
2 Instituto Nacional de Estadística Unidad Reajustable

laws. Internally, such a system could disrupt the current payment system, such as by moving money out of bank deposits.

The implementation of a national blockchain for payments would also require extensive investigation and, probably, a test project. While technologically, such a system could be implemented by copying an existing blockchain projects, most of those code bases have issues and have not be thoroughly reviewed.

Meristem

The Meristem effort[1] is as an experiment which aims to provide tools for blockchain experimentation. One aim is pedagogic, to build a code base able to generate a functional blockchain system which contains only the elements strictly necessary to its function. This research aims, first, to discover the minimal set of components necessary for a blockchain which can provide a given set of functions, second, to build a code base able to instantiate blockchain systems with variable functions, and third, to use the flexible code base to generate multiple blockchain systems and investigate inter-system interactions such as anchoring and sidechain mechanisms.

The experiment has not progressed very far. As a preliminary step, Meristem aims to be able to build two types of blockchain systems: one type which handles transactions with arbitrary (though perhaps size limited) content, another type providing a simple cryptocurrency. We aim to be able to record the current state of any one Meristem blockchain as a declaration sent to another Meristem blockchain. The first proof of concept would be a setup running several chains, each using independent nodes, with all the chains anchored on a chain allowing arbitrary declarations.

Meristem is a toy system, absolutely unsuitable for use! Meristem is written in Javascript, whose dynamic typing makes it unsuitable for the rigour required by actual blockchain systems. Javascript serves a pedagogic goal: it provides analytic transparency for clarity and offers a rapid prototyping platform for development. Meristem abandons all formal rigour such as considering formally the byte structure of data and messages. This implementation is not cryptographically secure and has no formal binary definition. Meristem, in its current form, is suitable only for educational exploration.

1 Meristem

Appendix A. On Digital currency systems

The most popular use of blockchain asset tracking systems has been to exchange digital representations of monetary value, *i.e.* as currency systems.

> ***This discussion is <u>not</u> an endorsement of cryptocurrencies.***
>
> This discussion is a *technical presentation* of information systems which use blockchains for the ownership and exchange of provably scarce, digital resources. The question of whether these scarce digital resources actually have any value is *not* technical, but rather one of economic judgment.
>
> The use of blockchain projects for holding and trading currency appears to function. That the code work at all, that the information system (the code, servers, network, messaging) survive, that the community hold together, that the economy provide value to its participants are all testimony to the brilliance of Bitcoin. Satoshi Nakamoto, whomever that may be, deserves high praise indeed. What a hack!
>
> Which is not to say that Bitcoin is perfect, nor even *viable*. There are no clear reasons to assign any real world value to these digital representations. There are many reasons to suspect the whole system could could collapse. Many issues arise in the use of blockchain projects for holding and trading virtual currency: issues in the design of the systems, in their implementation through software code, in the viability of their communities, and in the structure of their economies, including their joint economy. This presentation seeks to inform readers to reach their own conclusions on the subject.

Digital currency systems involve many issues, structural, technical, economic, and legal. However, before entering into a discussion of these issues, we must first understand some fundamental differences in digital currency systems, and agree on some terminology.

Digital currency systems use similar mechanisms to manage two different types of assets: assets representing on-chain, digital resources, which we will call *cryptocurrencies*, or assets representing off-chain, real world, deposits of fiat[1] money (or precious metals), which we will call *crypto-fiat currencies*. These two asset types are treated, in large part, similarly by blockchain systems but have different characteristics.

Cryptocurrency assets either have no intrinsic value or have value specifically for the use of the blockchain system. Cryptocurrency assets depend only on the blockchain system for their generation, exchange, (and possibly retirement). Cryptocurrency assets require some mechanism to limit the amount generated; this mechanism essentially amounts to a *monetary policy*, in the sense used by economists. Cryptocurrency systems face regulatory uncertainty, beyond that faced by all digital currency systems and crypto-fiat currency systems, because the lack of intrinsic value opens the interpretation to these being 'securities'.

We must also divide cryptocurrency assets into two kinds: assets which interact with the blockchain systems, that is assets that the blockchain software recognizes and uses, as against assets which merely use the blockchain systems. Most blockchain systems require fees for transactions and for processing, both to pay for usage costs and to restrict spurious usage (*spam*). In such system, one

1 The term *fiat* refers to currency declared valid (by fiat) in a jurisdiction, and therefore have intrinsic value at least as a way to pay taxes in that jurisdiction.

cryptocurrency asset is usually privileged in that it serves to pay these fees. The Bitcoin system requires payment transaction fee in the *Bitcoin* currency, making that privileged asset. The Ethereum system requires *Ethereum* payments for transaction fees and *gas* payments for processing fees, but the latter is purchased with the former so the *Ethereum* asset is privileged. These privileged, intrinsic cryptocurrency assets we will call *intrinsic coins*. In contrast, other digital assets have no connection to the underlying system. The myriad of ERC-20 tokens issued on the Ethereum platform all run on the system without interacting with it directly. We will call these unprivileged, extraneous cryptocurrency assets, *tokens*.

Crypto-fiat currency assets, unlike cryptocurrency assets, derive their intrinsic value from the underlying fiat (or precious metal). Crypto-fiat currency assets depend on more than just the blockchain system; they require that some party provide the fiat backing for the digital resource. The need for this *registrar* implies a new point of trust beyond the underlying blockchain system. The overall monetary policy depends principally on the monetary policy of the underlying fiat currency. Crypto-fiat currencies face legal issues related to the fiat backing.

Digital currency assets, then, are *cryptocurrencies*, either *intrinsic coins* or *tokens*, or are *crypto-fiat currencies*.

Currency systems necessarily require *trust*; users must believe the currency instrument has value which can be realized during the transfer of the instrument to some other owner. While cryptocurrencies were created to avoid having to trust third parties such as governments and banks, in actuality cryptocurrency systems merely shifted the required trust. Digital currency systems depend on trusting the blockchain system, trusting the community building and running the system, and trusting oneself.

Users of blockchain based, digital currency systems must trust the blockchain system itself. Users must trust the design and implementation of software code at all levels, the cryptographic library code, the blockchain processing code, the node networking code and the digital currency asset management code. Users must be confident that the asset management code fully controls the creation, transfer, and possibly of destruction, of currency assets. In cryptocurrency systems, users must trust that the monetary policy implicit in the creation of assets is essentially fair to all participants. In crypto-fiat currencies, users must trust the registration system for the fiat assets including trusting the registrar, the backer with fiat, and the fiat storage system.

Users of blockchain based, digital currency systems must trust the community running the blockchain system. Users must trust the community can survive, that the community can maintain the running system and pay its costs. Users must further trust the community not to alter the essentials of the system code nor to alter the blockchain itself and its history of past events.

Users of blockchain based, digital currency systems also must trust themselves to understand the system completely, to monitor it, and to use it always following best practices. This requires much more trust in oneself than when using cash, for example.

Blockchain based, digital currency systems face many technical issues: in scaling the systems, in maintaining a sane ecological footprint for the system, and in valuing the currency.

Current blockchain projects generally cannot handle the sustained number of transactions required of a global payment system. Many projects are exploring solutions to scaling in many different directions.

The current electrical consumption of the Bitcoin network is embarrassing. The intentional inefficiency of the proof-of-work puzzles exists only because hashing is so trivial and fast. Instead, many projects are moving to proof-of-stake or to other mechanisms which do not consume such ridiculous amounts of energy.

Currency systems function within a larger economic system, whose structure determines the value of the currency instrument and whose monetary policy affects the usage of the instrument. Crypto-fiat currencies reflect the underlying fiat currency, both for the value of the instrument and for the monetary policy of that fiat. In contrast, cryptocurrencies require their own valuation and monetary policy.

The value of a currency instrument can be estimated through multiple, alternative approaches, based on intrinsic value, utility value, market value, cost of generation, or expected future value. Cryptocurrency instruments either have no intrinsic value, as for tokens, or have an intrinsic value based on their utility as payment for the service of the blockchain system, as in intrinsic coins. A utility approach to valuation considers the value of the asset to be a function of its usage. This approach usually falls back on the equation of exchange[1]:

$$\text{Money Supply} \cdot \text{Velocity} = \text{Price} \cdot \text{Quantity}$$

although, when actually used, one term usually absorbs the errors in the estimates[2]. The market value of cryptocurrency instruments is generally well defined, at least for those actively traded on exchanges, reflecting the balance between offer for sale and demand for purchase; however, market value is subject to the whims of the market and to manipulations of market participants[3,4,5,6]. The cost of generation of cryptocurrencies, usually related to the cost of computational machines and of electricity[7], sets a base cost which must be obtained for generation to remain viable[8]. The expected future value is harder to define due to the volatility in the market value, although the arrival of futures trading provides a better defined future valuation.

The actual value assigned to an instrument will affect its economic function, regardless of the specific source of valuation. Any instrument with zero valuation can not serve as a currency; conversely, any non-zero valuation means the instrument can act as a currency. An instrument with increasing value will tend to favour holding or hoarding the instrument rather than using it; conversely, a falling value will favour spending the instrument as quickly as possible, possibly leading to currency runs and a crash in value. If the value of an instrument is not stable, the instrument can not serve as an instrument for lending, also meaning that there is no common rate of interest for the instrument, greatly restricting the instrument's utility in an economy.

1 Mike Sall Valuing Cryptoassets from the Ground Up 2018-04-23
2 Alex Evans On Value, Velocity and Monetary Theory 2018-01-28
3 Oscar Williams-Grut Meet the crypto trader ... 'pump and dump' profits... *Business Insider* 2017-12-08
4 John Griffin and Amin Shams Is Bitcoin really Un-Tethered? 2018-06-25
5 Matt Robinson and Tom Schoenberg U.S. Launches Criminal Probe into Bitcoin Price Manipulation *Bloomberg* 2018-05-24
6 Frances Coppola Cryptocurrency Trader Says The Market Is Manipulated *Forbes* 2018-06-01
7 Timothy Lee Bitcoin's insane energy consmption, explained *Ars Technica* 2017-12-06
8 Gregory Trubetskoy Electricity cost of 1 Bitcoin (Sep 2017) 2017-09-28

Monetary policy determines the economic function of the monetary instrument. In cryptocurrencies, monetary policy establishes the amount of currency in circulation, the schedule under which new currency issued into circulation (or withdrawn from circulation), and the allocation strategy for the currency. In blockchain systems, the monetary policy is usually fixed. For example, the policy for Bitcoin is for issuance of new coins with every generated block of the blockchain leading to a final total of 21 million Bitcoins. When the currency in circulation is fixed, the system tends to be deflationary, increasing the value of the currency instrument; alternatively, if new currency is constantly introduced, the economic system tends to be inflationary, lowering the value of the currency. Issuance of the currency can occur when the currency is created, can be progressive, or can happen at certain moments. The allocation of new currency can fall to founders of a blockchain system, to a foundation in charge of the system, to operators running the system, or to existing owners of currency. No cryptocurrency system has yet established a dynamic monetary policy able to respond to market conditions to manipulate the system as is done by central banks in national currencies.

The legal status of community run, trans-national digital currencies has always been problematic. These currencies were explicitly designed to move outside the direct reach of the legal system. However, the legal systems of the different nations has been working systematically to tame these currency systems through rules which affect individuals, rules which affect the collective computation, and rules which affect businesses interfacing between the outside world and the blockchain projects. The legal status of these digital currencies is still being invented.

Appendix B. Blockchain Projects

A few blockchain related projects deserve special focus. Here we examine four of the most important: the Bitcoin, Ethereum, and Cardano blockchain projects and the Hyperledger free software project.

Bitcoin

The Bitcoin blockchain project is the canonical first generation blockchain system. It had the unique goal of providing a digital currency which could be traded directly between end users without needing to trust an external third party.

The release in 2008 of the paper *Bitcoin: A Peer-to-Peer Electronic Cash System* by Satoshi Nakamoto and of the bitcoin software in 2009, marked the beginning of viable, blockchain based cryptocurrencies and launched the world of "blockchain."

Beyond simply being the first project based on a shared blockchain, Bitcoin brilliantly solved many of the core issues facing a digital currency which eschewed any trusted authority. Bitcoin's most important contribution was to show that a digital currency system was possible. A digital currency system must be able to produce artificially scarce digital resources, to establish ownership of these resources, to place solely with the owners the ability to transfer those resources, and to ensure that the owners only transfer the resources once. Bitcoin solved these challenges using cryptographic primitives and a network of participants. Provably scarce resources were to be produced through luck, assigned to the first participant able to solve a probabilistically difficult puzzle. Ownership was solved using account addresses derived from the public keys of cryptographic key pairs. Control of the transfer of resources was solved using cryptographic signatures based on those key pairs. Finally, the inability of owners to duplicate transfers was solved by inverting the usual trust model and sharing publicly the history of all transactions. Beyond this purely technical solution, Bitcoin also struck a remarkable balance of interests between its participants. Participation was completely unrestricted so that everyone could use in the system, both as resource owners or as resource producers. The balance in the economic design of the system provided room for new participants through a growing (but eventually finite) money supply assigned probabilistically by amount of effort contributed. Bitcoin's brilliant design ensured its ongoing popularity, the ongoing popularity ensured that it would be studied, and it study ensure its use as inspiration for other projects.

One of the aspects of *Bitcoin*'s success is its simplicity. The *Bitcoin* system had a single focus on the provision of a cryptocurrency and developed a minimalistic solution to that aim. This simplicity makes *Bitcoin* useful as an entry point for the study of blockchain projects in general.

Another aspect of Bitcoin's success lies in its openness: anyone can participate as a user or as a node on the network.

Another aspect of Bitcoin's success comes from its economic structure: a well defined monetary policy and a distributed incentive structure.

Bitcoin References

https://bitcoin.org
https://github.com/bitcoin/bitcoin/

https://lists.linuxfoundation.org/mailman/listinfo/bitcoin-dev
https://www.reddit.com/r/Bitcoin

Bitcoin Developer Guide https://bitcoin.org/en/developer-guide
Bitcoin Developer Reference https://github.com/minimum/Bitcoin-Spec

Satoshi Nakamoto *Bitcoin: A Peer-to-Peer Electronic Cash System* 2008

Blockchain Info https://blockchain.info

Books:
Pedro Franco *Understanding Bitcoin: Cryptocurrency, Engineering, and Economics* John Wiley & Sons, 2014
Arvind Narayanan *et al*., *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction* Princeton University Press, 2016
Andreas Antonopoulos *Mastering Bitcoin* O'Reilly Media, Inc. 2017

Articles:
Davide De Rosa *A developer oriented series about Bitcoin* 2005 https://daviderosa.com/basic-blockchain-programming/

Videos:
Andreas Antonopoulos *Andreas M. Antonopoulos: Advanced Bitcoin Scripting* (YouTube video)

Ethereum

The Ethereum project serves as the canonical Blockchain 2.0 project. Ethereum was undertaken explicitly to extend the capabilities of a blockchain project beyond what was supported by the Bitcoin project, most notably in extending the capabilities of the virtual machine for the scripting language. The core technological goal of Ethereum developers was to have a blockchain system able to process arbitrarily complex scripts and even to store such scripts for future invocation.

The release of the "white paper" by Vitalik Buterin in late 2013 and of the "yellow paper" by Dr. Gavin Wood in 2014 launched the *Ethereum* project intending to develop a more flexible alternative to *Bitcoin*. In the middle of 2014, a crowd sale of the future Ether tokens managed to raise around $18 million funds for the project. The successful launch in July 2015 created a space for the rapid expansion on projects using the Ethereum blockchain and its Ether currency.

The first major stored procedure designed to act as an autonomous organization collected a vast amount of funds but, before it could use those funds for innovation, was hacked by an outsider. In response to this massive theft, the community split into two parts, one part taking on the name *Ethereum Classic* and keeping the original rules while the other part kept the name Ethereum but altered the rules of the project to pay back the stolen funds.

Ethereum References

https://www.ethereum.org/
https://www.ethereum.org/foundation

Vitalik Buterin *et al*. *A Next-Generation Smart Contract and Decentralized Application Platform* (The Ethereum White Paper) https://github.com/ethereum/wiki/wiki/White-Paper

Gavin Wood *Ethereum: A Secure Decentralized Generalized Transaction Ledger* 2014-Present
https://ethereum.github.io/yellowpaper/paper.pdf

Developer source code: https://github.com/ethereum/
Developer documentation: http://www.ethdocs.org/en/latest/
Truffle development Framework: https://github.com/trufflesuite/

A blockchain explorer: https://www.etherchain.org/

Articles:
Wikipedia "*Ethereum*" https://en.wikipedia.org/wiki/Ethereum
Taylor Gerring *Cut and try: building a dream* 2016 https://blog.ethereum.org/2016/02/09/cut-and-try-building-a-dream/
Various *History of Ethereum* 2016-Present
https://ethereum-homestead.readthedocs.io/en/latest/introduction/history-of-ethereum.html

https://dappsforbeginners.wordpress.com/
https://ethereumbuilders.gitbooks.io/guide/en/index.html
https://solidity.readthedocs.io/
http://truffleframework.com/docs/

Videos:

Gavin Wood *DEVCON1: Ethereum for dummies - Dr. Gavin Wood* 2015 YouTube https://www.youtube.com/watch?v=U_LK0t_qaPo
Vitalik Buterin *Devcon2: Ethereum in 25 Minutes* 2016 YouTube https://www.youtube.com/watch?v=66SaEDzlmP4
Vitalik Buterin *A Modest Proposal for Ethereum 2.0* 2017 YouTube https://www.youtube.com/watch?v=hAhUfCjjkXc
Vitalik Buterin *Ethereum in 25 Minutes*, Version MMXVII 2017 YouTube https://www.youtube.com/watch?v=mCzyDLanA7s
Ethereum Foundation's channel on YouTube https://www.youtube.com/channel/UCNOfzGXD_C9YMYmnefmPH0g
DesignCourse *Developing Ethereum Smart Contracts for Beginners* 2017 https://www.youtube.com/watch?v=KU6bvciWgRE&list=PL0lNJEnwfVVMuX2Ds19Wj_7Mcze3FDJr3
Siraj Raval *Ethereum Explained* 2017 YouTube https://www.youtube.com/watch?v=-_Qs0XdPpw8
Fabian Vogelsteller *Web3.js 1.0* 2017 YouTube https://www.youtube.com/watch?v=92pdrRH_VGA

Cardano

The Cardano project is a good example of a third generation blockchain project. Cardano aims to provide all the functionality of Ethereum, even using the same virtual machine for script execution, but adopts more formal development practices to ensure code correctness. One of the founders of the project, Charles Hoskinson, presents the challenge of the project as adopting formal methods and academic publications, as tackling scaling, self-governance, and interoperability with other systems including the traditional financial network.

Cardano References

https://cardano.org
https://cardanofoundation.org/
https://cardanodocs.com

https://github.com/input-output-hk/cardano-sl

Aggelos Kiayias *et al. Ouroboros: A Probably Secure Proof-of-Stake Blockchain Protocol* 2017
https://eprint.iacr.org/2016/889.pdf

Articles:
Wikipedia "*Cardano (platform)*" https://en.wikipedia.org/wiki/Cardano_(platform)

Videos:
Charles Hoskinson *IOHK | Cardano whiteboard; overview with Charles Hoskinson* 2017 YouTube
https://www.youtube.com/watch?v=Ja9D0kpksxw
Charles Hoskinson *Cardano (ADA coin) Founder Interview: Charles Hoskinson* 2018 YouTube
https://www.youtube.com/watch?v=pcNDSSmkgkA
Charles Hoskinson *IOHK | Charles Hoskinson: Third-Generation Blockchains* 2018 YouTube
https://www.youtube.com/watch?v=D_QLfnucgh0
Aggelos Kiayias *Ouroboros A Provably Secure Proof of Stake Protocol* 2017 YouTube
https://www.youtube.com/watch?v=fBKCbhX-dXI
Dionysis Zindros *IOHK | Cardano whiteboard; Sidechains, Dionysis Zindros.* 2018 YouTube
https://www.youtube.com/watch?v=04D2BP33YI8

Hyperledger

The Hyperledger project serves as a good example of the current generation of free software efforts aiming to provide the tools for the creation of custom blockchain networks. Hyperledger operates as a project of the Linux Foundation and hosts a number of separate software projects focused on blockchain technology. These include software projects providing implementations of blockchains, each with a slightly different focus., which can be used to implement specific blockchain networks for specific business communities. Hyperledger also hosts software projects which provide tooling for blockchain projects.

The general goals of the Hyperledger project, as presented on the website, are:

- Create enterprise grade, open source, distributed ledger frameworks and code bases to support business transactions
- Provide neutral, open, and community-driven infrastructure supported by technical and business governance
- Build technical communities to develop blockchain and shared ledger POCs, use cases, field trails and deployments
- Educate the public about the market opportunity for blockchain technology
- Promote our community of communities taking a toolkit approach with many platforms and frameworks

These goals aim to provide high quality software which can be used by communities that want to start a collaboration using a shared, distributed ledger as their official record.

The Hyperledger Architecture Working Group is attempting to establish, retroactively, a shared design philosophy to unite the disparate efforts. This group stresses the idea of layered, modular architectures so that each project can provide flexibility to its users. For example, the working group argues that all blockchain projects should be able to switch the consensus mechanism used by any particular implementation since the tradeoffs of any particular consensus mechanism might not be suitable for a given implementation.

The Hyperledger project includes a number of software efforts providing blockchain frameworks:

- *Burrow* is a permissionable smart contract machine built in part to the specification of the Ethereum Virtual Machine (EVM).
- *Fabric* is a permissioned system which allows both access control and use policy control, an architecture based on an execute-order-validate flow for transactions for scalability and privacy of transaction content, a modular platform allowing alternative consensus mechanisms, written in the language go and allowing smart contracts in multiple languages.
- *Indy* is a distributed ledger, purpose-built for decentralized identity.
- *Iroha* aims to be simple and easy to incorporate into projects requiring distributed ledgers.
- *Sawtooth* is a modular platform for distributed ledgers using a Proof of Elapsed Time consensus algorithm to minimize resource consumption.

and a number of software efforts providing tooling:

- *Caliper* is a blockchain benchmark tool.
- *Cello* provides deployment tooling for blockchains.
- *Composer* provides a domain specific language for building the business logic of blockchains.
- *Explorer* enables the examination of blockchain ledgers.
- *Quilt* implements the InterLedger Protocol ILP to transfer assets between ledgers.

This section focuses on the *Hyperledger Fabric* software project. The Fabric software project is mature and widely used by large companies, including IBM, Microsoft and others, providing blockchain projects on the cloud. Fabric enables permission control both through access control

requirements to use the system or run a node in the network and through a policy system which restricts allowable transactions. Fabric uses an 'execute-order-validate' transaction flow in which transaction execution happens separately from consensus ordering enabling better overall scalability and allowing the isolation of transaction contents to specific parts of the network.

This section also discusses the *Hyperledger Composer* tool, which enables the rapid creation of the business elements used in a blockchain system. Composer provides a domain specific language to develop the data model, scripts, access control rules, and queries used by a blockchain project. Composer currently runs these elements as an interpreted system on top of a Hyperledger Fabric system. Composer also provides tools for the rapid generation of software components usable for the creation of HTML interfaces.

Hyperledger References

https://www.hyperledger.org
https://www.linuxfoundation.org/
https://www.hyperledger.org/projects/fabric
http://hyperledger-fabric.readthedocs.io/en/latest/
https://www.hyperledger.org/projects/composer
https://blockchaindevelop.mybluemix.net/editor

Articles:

Akarsh Agarwal *Installing Hyperledger Fabric v1.0 on Ubuntu 16.04 - Part I/II/III* 2017 https://medium.com/mlg-blockchain-consulting/installing-hyperledger-fabric-v1-0-on-ubuntu-16-04-part-i-18043bde92bb
Akarsh Agarwal *Understanding the Node-SDK of Hyperledger and Running the First Example* 2017 https://medium.com/mlg-blockchain-consulting/understanding-the-node-sdk-of-hyperledger-and-running-the-first-example-5f6753393ec8
Eli Androulaki *et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains* 2018 https://arxiv.org/pdf/1801.10228v1.pdf
The Hyperledger Project H*yperledger Architecture, Volume 1 Introduction to Hyperledger Business Blockchain Design Philosophy and Consensus* 2018 https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf
The Hyperledger Project H*yperledger Architecture, Volume 2 Smart Contracts* 2018 https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf

Videos:

Nikolay Vlasov *What's new in Hyperledger Fabric v1.0* 2017 YouTube https://www.youtube.com/watch?v=6nGIptzBZis
Simon Stone *Introduction to Fabric Composer* 2017 YouTube https://www.youtube.com/watch?v=fdFUsrsv5iw
Simon Stone *Building Blockchain Apps for Node.JS developers With Hyperledger Composer* 2018 YouTube https://www.youtube.com/watch?v=gAxK6zYrfxI

Hyperledger Fabric History

Hyperledger Fabric emerged recently in the history of blockchains, with version 1.0 released in the middle of 2017. Fabric belongs to the generation of blockchain efforts which intend to provide the tools to build blockchain systems for specific use cases; it is not focused on building any working system for any particular use case such as enabling a global, digital currency. These systems are

therefore concerned with allowing flexibility, for instance allowing the integration of multiple assets rather than focusing on a single, privileged currency.

Hyperledger Composer shares this same intent of facilitating the building of bespoke systems. The Composer project attempts to solve the issue that developers were being lost in the code details rather than being able to focus on the business use case in which they were interested. The Composer tool enables developers to focus on the components in their use case, leaving for later the details of building the functioning, distributed network. Composer structures development work into three phases. First, developers define the abstract components of the system, including the data model of 'participants', 'assets', and 'transactions,' the access control rules, and the policy rules for transactions. Second, developers instantiate specific instances of the participants and assign assets to those participants. Third, developers can simulate exchanges between participants and query the system based on exchange history. Composer is mostly a prototyping tool which provides a quick way for developer to develop a proof-of-concept to show the system could work.

Hyperledger Fabric Technical Foundations

The biggest technological contribution of the Hyperledger Fabric system comes from its novel '*execute-order-validate*' transaction flow rather than the more widely used '*order-execute*' flow used by Bitcoin, Ethereum, and most existing systems. This design separates network nodes into three, functionally separate, groups: client nodes, endorsing nodes, and ordering service nodes. Transactions originate with client nodes who send their requests to endorsing nodes. The endorsing nodes pre-calculate the transaction to obtain a result defining the changes to the system state caused by the transaction, then generate an endorsement signature for this transaction, and finally bundle up the three. If the system requires more than one endorsement, this step may be repeated on other endorsement nodes resulting in more bundles. Next, the set of bundles are sent to the ordering nodes who, through some consensus mechanism, agree to include the transaction in a block at some particular place in the ordered sequence of transactions. Next, the transaction order is broadcast to the network of endorsing peers, each of which validates the transaction; it must follows the policy rules and must not conflict with any previous transaction in the sequence of the current block. Finally, the node registers the transaction in its own copy of the ledger either as failed or as successful and, if it is successful, also applies the changes calculated in the initial execution to the validating node's own model of the current system state. This latter process is fully deterministic.

The contribution of this approach comes from the decoupling of the execution of a transaction from the acceptance of that transaction into the history of the ledger. This separation enables better scaling since transaction execution, a potentially costly step allowing for arbitrarily complex logic, can happen in parallel to other execution and is decoupled from the necessarily sequential processing required for protection against concurrent double spend attempts. The separation also allows a measure of privacy since the details of the transaction are only known to the endorsement peers not to the whole network.

A second technological contribution of Hyperledger Fabric comes from its extension of the permission system from simple access control to include an integrated policy system. Permissioned blockchains restrict participation, either as a processing node or a client user, to known and validated users only. The policy system further restricts the system by requiring that known users of the system ensure that their transaction requests are endorsed by a specific group of nodes. The policy system is able to require that transactions have multiple endorsements or have endorsements

from specific peers. For example, a transaction between companies might be required to be endorsed by at least one endorsement node from each company. This policy system also involves the existence of special users called 'PeerAdmin' users which are allowed to change the policy rules of a system. The main contribution of this approach comes from it enabling the isolation of transaction details from the shared ledger and state, thereby allowing a measure of privacy in transactions.

A third technological contribution of Hyperledger Fabric arises from its use of 'channels' which are entirely separate blockchains. This approach enables a single network to handle multiple different blockchains.

A fourth technological contribution of Hyperledger Fabric comes from its modular design. The system is specifically designed so that any consensus system can be plugged in. Currently three consensus mechanisms can be used, one based on Kafka, one called 'solo' is a single instance used only for test networks, and one called 'BFT-SMaRt' which is new. The modularity of the design enables the use of a consensus mechanism appropriate for each specific implementation and eases innovation in the future.

A fifth technological contribution of Hyperledger Fabric arises from its extensive use of Docker containers. All peers, by default, are instantiated as stand-alone docker instances. Furthermore, all transactions are executed in a new docker instance setup uniquely for that transaction. The isolation provided by this use of containers enables the system to use arbitrary, well known computer languages for the code run during transactions rather than needing to restrict the system to a novel, bespoke language.

The complexity of this Hyperledger Fabric design, unfortunately, increases the difficulty to developers of developing functional networks. The Hyperledger Composer project exists partly to separate the work of defining the business logic from this complicated work building a functioning network.

The main technological contribution of Hyperledger Composer comes from its ability to run, via an interpretation layer, on top of a blockchain system. The result of defining the data model, transaction code, access control lists in its domain specific language can be converted on the fly to the rules needed by the blockchain system. This flexibility ensures that the domain specific knowledge required by Composer can serve even beyond a single blockchain system.

The ease of use which comes from the various Composer sub-projects, such as online playgrounds and automatically generated elements for web applications, leads the way forwards in the tooling that will be required of popular projects in the future.

Hyperledger Fabric Network

The 'execute-order-validate' system and the use of policies for validation lead to a network with an extra tier of nodes compared to most network. Since the permission system uses standard cryptographic certificates, most systems will have a Certificate Authority service. The processing system is separated into nodes of two types: endorsement nodes which calculate the consequence of transaction execution and endorse the transaction with a signature and order service nodes which use some consensus mechanism to agree on an ordered list of transactions to include in the next

block. All nodes then validate the transactions in the block based on deterministic rules of policy and non-conflict. Client nodes communicate with the endorsement nodes to find out the state of the system, to query the system, and to propose transactions.

Hyperledger Fabric Community and Surroundings

Since systems built with Hyperledger Fabric generally focus on supply chain networks tracking the transfer of non-monetary assets, these systems generally do not face regulatory and legal issues faced by cryptocurrencies. Projects using Fabric generally have a pre-defined set of participants who have agreed to use the system to help them track assets and therefore are already aware of the legal obligations they face.