Annals of Emerging Technologies in Computing (AETiC)
Vol. 3, No. 5, 2019

Review Article

Dining Philosophers, Byzantine Generals, and the Various Nodes, Users, and Citizens under Blockchain Rule

Denisa Reshef Kera

BISITE, University of Salamanca, Spain
denisa.kera@usal.es

Received: 13th November 2019; Accepted: 24th November 2019; Published: 15th December 2019

Abstract: Agreements, consensuses, protocols, resource-sharing, and fairness are all examples of social and political metaphors that define and shape new computational algorithms. The thought experiments and allegories about resource-sharing or agreement between nodes played a vital role in the development of "concurrent programming" (enabling processor power-sharing and process synchronization) and still later in the development of distributed computing (facilitating data access and synchronization). These paved the way for current concepts of consensus mechanisms, smart contracts, and other descriptions of cryptocurrencies, blockchain, distributed ledger, and hashgraph technologies, paradoxically reversing the relations between metaphor and artifact. New computing concepts and algorithmic processes, such as consensus mechanisms, trustless networks, and automated smart contracts or DAOs (Distributed Autonomous Organizations), aim to disrupt social contracts and political decision-making and replace economic, social, and political institutions (e.g., law, money, voting). Rather than something that needs a metaphor, algorithms are becoming the metaphor of good governance. Current fantasies of algorithmic governance exemplify this reversal of the role played by metaphors: they reduce all concepts of governance to automation and curtail opportunities for defining new computing challenges inspired by the original allegories, thought experiments, and metaphors. Especially now, when we are still learning how best to govern the transgressions and excesses of emerging distributed ledger technologies, productive relations between software and allegory, algorithms and metaphors, code and law are possible so long as they remain transitive. Against this tyranny of algorithms and technologies as metaphors and aspirational models of governance, we propose sandboxes and environments that allow stakeholders to combine prototyping with deliberation, algorithms with metaphors, codes with regulations.

Keywords: ***Blockchain; Bitcoin; Metaphors; Thought Experiments; Algorithms; Governance; Automation***

1. Introduction

Social and political metaphors played an essential role in the development of modern algorithms by defining the emerging computing and networking challenges in an engaging way. New

Denisa Reshef Kera, "Dining Philosophers, Byzantine Generals, and the Various Nodes, Users, and Citizens under Blockchain Rule", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 1-8, Vol. 3, No. 5, 15th December 2019, Published by International Association of Educators and Researchers (IAER), DOI: 10.33166/AETiC.2019.05.001, Available: http://aetic.theiaer.org/archive/v3/v3n5/p1.html.

algorithms that resolved issues of resource sharing between computer processes (concurrent computing), fault-tolerance, coordination and nodal synchronization (distributed computing) developed only after these issues had first gained popularity as thought experiments ("Dining Philosophers," "Two Generals" or "Byzantine Generals" problems) or metaphors (flows, catch/throw)[17]. These "unifying metaphors" and in the case of algorithms, stories, played a critical role in garnering support for and facilitating the adoption of new technologies by the general public, and in mobilizing the research community around a common goal leading to the development of new tools and technologies.

In his study of metaphors underpinning the development of parallel computing and transputers, Kari Leppälä [17] describes that during the "orientation phase of technology transfer," it is of the utmost importance "that a common and coherent understanding of characteristics and possibilities of the technology is created. A strong or commonly agreed concept enables researchers and engineers to perceive objects in [a] similar way and to concentrate efforts for solving problems or to create efficient technical artifacts. And this requires powerful metaphoric structures. "[17] In short, a good story or a metaphor mobilizes researchers insofar as it serves as a low-fidelity prototype that gives a clear challenge and a strong vision of some future product ("perceive objects in a similar way"), and in leaving open how to achieve it.

The utilization/dependence of metaphors in the "orientation" phase of the development of new technology (e.g., an algorithm) closely corresponds to the "conceptualization" role metaphors play in computing (also called "paradigm switches"), as discussed by Colburn and Shute [4]. They insist that the conceptualization function of metaphors goes beyond the pedagogical or cognitive roles. Metaphors make "sense when the agenda is primarily epistemological, that is when we are trying to account for new knowledge of existing reality (new ways of looking at cats and fog). But the subject matter of computer science is a product of its own creation. Transistors, computers, and programming languages do not exist prior to their conceptualization. Such conceptualization does not consist in new knowledge of old reality, but in new reality seeking a conceptual framework in which to be situated." [4] A good metaphor or story not only describes or reproduces something that already exists but fosters the development of a unique, new technology or algorithm.

When used in computational discourse, the principal function of metaphors is to simplify complexity rather than explaining something. They create a common "thread" among researchers allowing them to "reason about relatively concrete and very complex domains (like manipulating a runtime control stack) using the inferential structure of relatively simple and abstract domains (like catching and throwing)." [4] In this sense, the evolution of a metaphor, allegory or a thought experiment into an algorithm and later technology is neither random nor trivial but an essential component of innovation, and this is especially true in the case of emerging technologies and algorithms. Algorithms never serve as some solution to a riddle or clarification of a metaphor (story), but something generated and made possible by the story.

2. Algorithmic Governance as a Paradoxical Reversal of Metaphors and their Artefacts

While most studies on the use and role of metaphors in computing focus on their pedagogical and cognitive functions (to communicate the new artifacts to the public or promote mass adoption of the technology, such as better GUIs) [12,18,24,29,30], we are interested in the power of metaphors to mobilize communities of experts and generate new algorithms and technologies.

The generative capacity of good story and metaphor contributed to the development of important algorithms in the last half-century, and we claim that we are currently witnessing a paradoxical reversal of the vital relationship between algorithms and metaphors, stories and artifacts. Unlike the algorithms on which they were based, blockchain, distributed ledger, big data, AI and machine learning technologies transformed the relation to metaphors. They are nor products of challenging and provocative metaphors aimed at fostering the development of new algorithms and their adoption by the masses. Instead, these technologies are suddenly becoming a dominant metaphor for governance.

The powerful algorithms that originated from a set of famous thought experiments are reversing the causal order between a metaphor and its artifact. Instead of metaphors that guide the future development of technology, we have technologies and algorithms used as metaphors for algorithmic governance. This has profound implications on the meaning of governance, as well as on the variety of challenges posed to computing. It effectively whittles away at innovation, reducing governance and technology to their caricatured forms.

Although governance has historically been rooted in deliberation and negotiation, algorithms are reducing governance to automation, prioritizing the hunt for the right algorithm or technology over traditional modes of social interaction. Examples of these misuses of algorithms and technologies as metaphors of governance include "government policies driven and informed by prediction markets" in the case of Ethereum Founder Vitalik Buterin [3] embrace of Robin Hanson's concept of "Futarchy" [11] , "high-performance government' cognitive technologies'" inspired by security assessment and penetration testing developed by Dominic Cumming (the architect of Vote Leave), or calls for governments to be managed as a "lean startup" [31] by Tim O'Reilly (the founder and CEO of O'Reilly Media). In addition to these, there are different ideas of sandboxes and testnets that embody the cybernetic and technocratic dreams of "society-in-the-loop" [23] that couple policy with self-improving algorithms. How did the early algorithms that relied on metaphors and allegories of governance (such as food sharing and commensality) to define modern computing and networks become a model and metaphor for (algo)governance in their own right?

3. Dining Philosophers and Byzantine Generals Defining New Algorithms

The current algorithmic governance allegories reduce political processes to automated blockchain protocols, consensus mechanisms, and predictive analytics. The origins of the actual algorithms used in blockchain and computing are actually in two famous thought experiments about the "Dining Philosophers" [5] and later the "Byzantine Generals "[16]. These origin stories of modern computing demonstrate how random ideas about what it means to eat together and how to coordinate attack during a battle became powerful technologies that paradoxically claim to be the ultimate goal and model of governance and management of social phenomena.

The 1965 "Dining Philosophers" problem formulated by Edsger Dijkstra [5] was a response to the early time-sharing problem in computing, namely: how to program different processes or the peripherals so that they could "share" a common processor. It inspired early experiments with what is now called "concurrent" programming, algorithms that have come to define modern computing, as well as, solutions to synchronization and coordination issues in distributed computing networks. The issues with synchronization and coordination of processes, nodes, and components in a computer or a network strangely changed in the late 1970s from food sharing problems to fighting battles and war metaphors. The 1975 thought experiment that defined distributed computing algorithms made use of a gangsters scenario [1] and later, the "Two Generals" problem [10]. These directly led to the quite famous 1982 "Byzantine Generals Problem" [16] defining a functional network with unreliable and faulty nodes, the basis for all blockchain protocols.

In the late 1960s, commensality metaphors concerned with food and drink sharing practices were used mainly for pedagogical purposes. Foundational concurrent programming algorithms for early computer peripheries (mouse, screen, input, and output devices) and processes (software) emerged from a thought experiment that defined the problem of communicating and processing as coordination of talking and eating. The computer peripheries and software are like philosophers with strange eating habits that need to coordinate their talking and eating during a formal and ritual dinner. In the 1970s, these gustatory philosophers and festivities were swept aside, in favor of gangsters, generals, battlefield attacks and retreat scenarios in need of specific, novel algorithms capable of coordinating amongst unreliable nodes in new infrastructure (network). These later network experiments defined new potential modes of war coordination between (mis)trusting nodes.

Were these thought experiments completely random or simply by-products of a Cold War mentality obsessed with game theory? Would the results of these thought experiments (algorithms) still look the same if they had instead pushed forward other aspects of food commensality, such as hospitality to strangers, social bonding, socialization through rules and hierarchies, ceremonial communion? Or, instead of the battlefield metaphor, a dance performance paired with a close reading of Carl von Clausewitz on the importance of chance and uncertainty in war? Is the current dearth of interesting thought experiments and metaphors a sign of technological stagnation rather than maturity, and perhaps even cause for concern?

4. Blockchain Governance of Nodes, Users, and Citizens

The current algorithmic governance allegories reduce political processes to automated blockchain protocols, consensus mechanisms, and predictive analytics. The origins of the actual algorithms used in blockchain and computing are actually in two famous thought experiments about the "Dining Philosophers" [5] and later the "Byzantine Generals "[16]. These origin stories of modern computing demonstrate how random ideas about what it means to eat together and how to coordinate attack during a battle became powerful technologies that paradoxically claim to be the ultimate goal and model of governance and management of social phenomena.

The 1965 "Dining Philosophers" problem formulated by Edsger Dijkstra [5] was a response to the early time-sharing problem in computing, namely: how to program different processes or the peripherals so that they could "share" a common processor. It inspired early experiments with what is now called "concurrent" programming, algorithms that have come to define modern computing, as well as, solutions to synchronization and coordination issues in distributed computing networks. The issues with synchronization and coordination of processes, nodes, and components in a computer or a network strangely changed in the late 1970s from food sharing problems to fighting battles and war metaphors. The 1975 thought experiment that defined distributed computing algorithms made use of a gangsters scenario [1] and later, the "Two Generals" problem [10]. These directly led to the quite famous 1982 "Byzantine Generals Problem" [16] defining a functional network with unreliable and faulty nodes, the basis for all blockchain protocols.

In the late 1960s, commensality metaphors concerned with food and drink sharing practices were used mainly for pedagogical purposes. Foundational concurrent programming algorithms for early computer peripheries (mouse, screen, input, and output devices) and processes (software) emerged from a thought experiment that defined the problem of communicating and processing as coordination of talking and eating. The computer peripheries and software are like philosophers with strange eating habits that need to coordinate their talking and eating during a formal and ritual dinner. In the 1970s, these gustatory philosophers and festivities were swept aside, in favor of gangsters, generals, battlefield attacks and retreat scenarios in need of specific, novel algorithms capable of coordinating amongst unreliable nodes in new infrastructure (network). These later network experiments defined new potential modes of war coordination between (mis)trusting nodes.

Were these thought experiments completely random or simply by-products of a Cold War mentality obsessed with game theory? Would the results of these thought experiments (algorithms) still look the same if they had instead pushed forward other aspects of food commensality, such as hospitality to strangers, social bonding, socialization through rules and hierarchies, ceremonial communion? Or, instead of the battlefield metaphor, a dance performance paired with a close reading of Carl von Clausewitz on the importance of chance and uncertainty in war? Is the current dearth of interesting thought experiments and metaphors a sign of technological stagnation rather than maturity, and perhaps even cause for concern.

5. From Metaphors and Thought Experiments to Allegories of Algorithms

The early computing thought experiments that yielded some of the most innovative and ground-breaking algorithms and technologies were responses to metaphors of governance and social interactions intentionally lacking historico-political accuracy and significance. This prevented the confusion we are witnessing today as simplified views of governance masquerade as computing problems and underly current attempts to replace governance with the right consensus algorithm.

By way of example, the "Dining Philosophers" problem reduced the lush anthropological concept of commensality to its respective biological and philosophical components of eating and thinking, and because the scenario was so extreme, no one ever thought to use such algorithms for actual dinners. The problem was employed to help students understand computational processing and communications issues, not in order to explain dining to future programmers or propose an algorithm that would regulate feeding or change the eating culture.

All this changed in 1982, with Leslie Lamport's influential thought experiment, the "Byzantine Generals Problem" that purports to have some historical accuracy. This tendency for more literal use of allegories is even more visible in his later use of the Greek island of Paxos and its parliament to describe a distributed computing algorithm. While the original Byzantine Generals text used a known "figure of speech" in computing (the metaphor of the generals) to define a simple thought experiment about a battle, the Paxos algorithm is a complex allegory pretending to describe a possible parliament even if it is just a parody with no intention to reduce governance to the Paxos algorithm.

Lamport's original 1970s algorithms for distributed networks with faulty, unreliable, or even malicious peers played an essential role in the development of cryptography (i.e., digital signature) and later blockchain technologies (i.e., consensus mechanisms). It also opened Pandora's box ushering in a wave of algorithmic governance metaphors. According to Lamport's own account, the goals of his famous thought experiment and the later less famous Paxos allegory were neither pedagogical nor exploratory. He had simply wanted to popularize the algorithmic challenge and amuse his readers. He explicitly mentions the popularity of the dining philosophers problem as his motivation to invent a story that would similarly attract the attention of his peers: "I have long felt that, because it was posed as a cute problem about philosophers seated around a table, Dijkstra's dining philosopher's problem received much more attention than it deserves. (For example, it has probably received more attention in the theory community than the readers/writers problem, which illustrates the same principles and has much more practical importance." [32]

In the case of the Paxos algorithm, eager to reproduce the success he had enjoyed with the generals, Lamport decided to describe a lost parliament on an ancient Greek island by way of an elaborate allegory with details rather than a simple thought experiment. He went to such extremes that he named the ancient Greek legislators in his allegory after known computer scientists working in the field of distributed computing, and even dressed as an Indiana Jones archaeologist during a few of his lectures. Despite his best intentions and whole-hearted performance, his allegorical initiative was a failure: "My attempt at inserting some humor into the subject was a dismal failure. People who attended my lecture remembered Indiana Jones, but not the algorithm. People reading the paper apparently got so distracted by the Greek parable that they didn't understand the algorithm. "[15] The paper was rejected in 1990, but eventually published in 1998, and finally won a prize only in 2012.

The failure of the Paxos allegory, as compared to the success of the Byzantine Generals thought experiment, can be attributed to its over-abundance of details that too closely emulated reality and distracted the readers from the "moral" of the story which was also too literal (it was a detailed description of an algorithms solving a puzzle rather than a provocation that inspires readers to come up with new solutions). Unlike previous metaphors and thought experiments that incited users to come up with new ideas (e.g., algorithms), Lamport's allegorical algorithm was a high-fidelity prototype so complete and concrete that it left little room for feedback or imagination and proved too complicated to communicate its basic underlying ideas and principles.

6. The Algorithmic Pamphlet that announced Algorithmic Governance

The example of the Paxos allegory demonstrates why algorithms should not parallel reality (however imaginary or historical) too strictly: with little to no room for further discussions and development, algorithms come to define and shape reality. It is this problematic tethering of algorithm and reality that underpins algorithmic governance and the whole idea of imbuing "governance-by-design" [20] with decision-making power over design principles, algorithms, code, technology, and ultimately society. This opens the floor to random, arbitrary, hidden, and "hazardous" decisions and choices made by the designers that "recede from the political as they become what 'is' rather than what politics has determined ought to be."[20]

This is particularly problematic in the case of blockchain algorithms and "smart contract" technologies that use allegories to drive forward the future of the world as a foregone conclusion for which algorithmic governance is the only possible governance solution. The blockchain algorithm (such as Bitcoin) became a literal model of novel forms of algorithmic governance. The actual blockchain even includes a pamphlet mocking the traditional forms of governance. Satoshi's famous hidden message in the Bitcoin 0 Genesis block is a newspaper headline that summed up the broken financial systems during the economic crisis: "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks" [33].

This algorithm of a new form of economic governance with a hidden pamphlet against traditional governance pushed even further that allegorical misuse of algorithms from the Paxos example. The pamphlet-backed algorithm not only reversed the logic of previous allegories, metaphors, and thought experiments that used governance issues to discuss and generate new algorithms but claimed destruction of the old models of governance. It explicitly calls for a change to governance by announcing that we are actually "entering an era of policymaking by 'design war'" [20]. The Bitcoin algorithm with a political pamphlet marks the final end of interesting metaphors used for computing and we are entering an era when existing algorithms are claiming a power over governance.

The idea of governance as an algorithm that promises a "technological fix" is problematic for numerous reasons. Such an approach lacks the "flexibility [required] to balance equities and adapt to changing circumstances" [20] or accommodate ambiguous and general values such as human rights ("privileges one or a few values while excluding other important ones, particularly broad human rights") [20], as well as legitimacy and transparency ("governance-by-design decisions that broadly affect the public are often made in private venues or in processes that make technological choices appear inevitable and apolitical") [20]. Most importantly, however, is that the reduction of governance to automation by way of algorithms curtails also innovation, it significantly diminishes the future development of algorithms. It actually closes both the future development of both, interesting new algorithms and governance models.

7. Conclusion: Future Governance Testnets and Sandboxes

The thought experiments that sought to create agreements in the presence of fault or uncertainty (fault-tolerance) and played such a vital role in the design of distributed computing algorithms (and the development of the blockchain protocols) were not principally problems of governance or war tactics and strategy. Unfortunately, the current fantasies of algorithmic governance modeled after these consensus mechanisms and protocols forget that they are based on two origin metaphors and they ignore other possible alternatives.They have reduced complex decision-making and deliberation processes (and issues) to pre-determined agreements on data values among multiple nodes that contain identical information and code. It is both politically naïve and problematic for individual citizens participating in such systems to be reduced and replicated as nodes performing the same code in strict accordance with the one prevailing interpretation of the network's values and rules. While smart contracts are self-executing and automated, legal contracts are always open for interpretation and even conflict that actually improves the legitimacy of the whole system.

Algorithms do not embody nor magically align with our values however consensus-driven they may be, like any other tool, they benefit those groups and individuals that best know how to use them to their advantage. Like electricity or the internet or any technology that becomes implemented on a mass scale and an indispensable part of our public infrastructure, they wield influence over our socioeconomic life and trigger traditional concerns over their fair and just distribution, the potential for their misuse, and implementation. Like the technologies that preceded them, as algorithms continue to generate new institutions, they will become increasingly subject to a compliance framework comprised of regulations, laws, institutional codes of conduct, ISO norms, and industry standards.

The real challenge at present is to involve various stakeholders in defining and implementing these standards and rules for this new infrastructure. We need to prioritize tools and environments that enable everyone to understand the new infrastructure, its possible future uses, and how it converges with existing technologies. This understanding will enable stakeholders to articulate their expectations, define metrics, and negotiate possible conflicts.

One such model proposes the creation of policy sandboxes [2,6,19] and "testnets" for experiments that involve the relevant stakeholders but excluding the general public. The solution we are currently testing is a type of "sandbox," or simulation tool for testing scenarios, that uses functional prototypes of blockchain services to explore various regulatory forms in a participatory setting [13,14]. These governance sandboxes and testnets are poetic machines that enable citizens to test how metaphors become algorithms and vice versa, allowing us to reflect upon the experience of what it means to live on such platforms and networks and how best to improve them.

References

[1] E A Akkoyunlu, K Ekanadham, and R V Hubert. Some Constraints and Tradeoffs in The Design of Network Communications.

[2] Lev Bromberg, Andrew Godwin, and Ian Ramsay. 2017. Fintech sandboxes: Achieving a balance between regulation and innovation. *Journal of Banking and Finance Law and Practice*.

[3] Vitalik Buterin. 2014. *An Introduction to Futarchy*.

[4] T. R. Colburn and G. M. Shute. 2008. Metaphor in computer science. *Journal of Applied Logic* 6, 4: 526–533.

[5] E. W. Dijkstra. 1971. Hierarchical ordering of sequential processes. *Acta Informatica* 1, 2: 115–138.

[6] Pei Sai Fan. 2017. Singapore Approach to Develop and Regulate FinTech. In *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 1: Cryptocurrency, FinTech, InsurTech, and Regulation*.

[7] Primavera De Filippi and Samer Hassan. *Blockchain Technology as a Regulatory Technology From Code is Law to Law is Code*.

[8] Primavera De Filippi and Benjamin Loveluck. 2016. The invisible politics of Bitcoin: governance crisis of a decentralised infrastructure.

[9] Primavera De Filippi and Internet Policy Review. 2014. Bitcoin: a regulatory nightmare to a libertarian dream.

[10] J. N. Gray. 1978. Notes on data base operating systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 393–481.

[11] Robin Hanson. 2007. The Policy Analysis Market (A Thwarted Experiment in the Use of Prediction Markets for Public Policy). *Innovations: Technology, Governance, Globalization* 2, 3: 73–88.

[12] Sattar Izwaini. *A corpus-based study of metaphor in information technology*.

[13] Denisa Reshef Kera. 2020. Anticipatory policy as a design challenge: Experiments with stakeholders engagement in blockchain and distributed ledger technologies (bdlts). *Advances in Intelligent Systems and Computing*, Springer Verlag, 87–92.

[14] Denisa Reshef Kera, Petr Šourek, Mateusz Kraiński, Yair Reshef, Juan Manuel Corchado Rodríguez, and Iva Magdalena Knobloch. 2019. Lithopia. *Extended Abstracts of the 2019 CHI Conference on Human Factors in*

Computing Systems - CHI '19, ACM Press, 1–6.

[15] Leslie Lamport. The Writings of Leslie Lamport. Retrieved November 13, 2019. Available: https://lamport.azurewebsites.net/pubs/pubs.html.

[16] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. *The Byzantine Generals Problem*.

[17] Kari Leppälä. *Diffusion of parallel computing and transputing concepts. A trial for unifying metaphors*.

[18] Carol G. Lombard. 2005. Conceptual metaphors in computer networking terminology. *Southern African Linguistics and Applied Language Studies* 23, 2: 177–185.

[19] Julie A. Maupin. 2017. *Mapping the Global Legal Landscape of Blockchain and Other Distributed Ledger Technologies*. .

[20] Deirdre K Mulligan and Kenneth A Bamberger. 2018. Saving governance-by-design. *California Law Review* 106, 697–784. Retrieved July 21, 2019. Available: https://doi.org/10.15779/Z38QN5ZB5H.

[21] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System.

[22] Satoshi Nakamoto. 2008. *Bitcoin: A Peer-to-Peer Electronic Cash System*.

[23] Iyad Rahwan. 2017. Society-in-the-Loop: Programming the Algorithmic Social Contract.

[24] Lee Ratzan. Making sense of the Web: a metaphorical approach.

[25] Nick Szabo. 1996. *Smart Contracts: Building Blocks for Digital Markets*.

[26] Nick Szabo. 1997. *Formalizing and Securing Relationships on Public Networks*.

[27] Nick Szabo. 1997. *The God Protocols*.

[28] Nick Szabo. 2002. *A Formal Language for Analyzing Contracts*.

[29] Eric Min-yang Wang and Alan Yu-hsing Huang. 2000. A Study on Basic Metaphors in Human-Computer Interaction. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 44, 1: 140–143.

[30] Jingfang Wu and Rong Chen. 2013. *Metaphors Ubiquitous in Computer and Internet Terminologies*.

[31] WTF is the Future of Tech? An Interview with Tim O'Reilly - Lean Startup Co. Retrieved November 13, 2019. Available: https://leanstartup.co/wtf-is-the-future-of-tech-an-interview-with-tim-oreilly/.

[32] The Byzantine Generals Problem - Microsoft Research. Retrieved November 13, 2019. Available: https://www.microsoft.com/en-us/research/publication/byzantine-generals-problem.

[33] Blockchain Explorer - Search the Blockchain | BTC | ETH | BCH. Retrieved November 13, 2019. Available: https://www.blockchain.com/btc/tx/4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b?show_adv=true.