2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

An Overview of Smart Contract: Architecture, Applications, and Future Trends

Shuai Wang[1,2], Yong Yuan*[1,3] (*Corresponding author*, *Senior Member*, *IEEE*), Xiao Wang[1,3], Juanjuan Li[1,3],
Rui Qin[1,3], Fei-Yue Wang[1,3,4](*Fellow*, *IEEE*)

[1]The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation,
Chinese Academy of Sciences, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]Qingdao Academy of Intelligent Industries, Qingdao 266109, China
[4]Research Center of Military Computational Experiments and Parallel Systems,
National University of Defense Technology, Changsha 410073, China
{wangshuai2015, yong.yuan, x.wang, juanjuan.li, rui.qin, feiyue.wang}@ia.ac.cn

***Abstract*— With the rapid development of cryptocurrency and its underlying blockchain technologies, platforms such as Ethereum and Hyperledger began to support various types of smart contracts. Smart contracts are computer protocols intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts have broad range of applications, such as financial services, prediction markets and Internet of Things (IoT), etc. However, there are still many challenges such as security issues and privacy disclosure that await future research. In this paper, we present a comprehensive overview on blockchain powered smart contracts. First, we give a systematic introduction for smart contracts, including the basic framework, operating mechanisms, platforms and programming languages. Second, application scenarios and existing challenges are discussed. Finally, we describe the recent advances of smart contract and present its future development trends, e.g., parallel blockchain. This paper is aimed at providing helpful guidance and reference for future research efforts.**

Keywords—smart contract; Ethereum; ACP approach; parallel blockchain

I. Introduction

In recent years, the development of blockchain technology has enabled customizable programming logic to be stored in a decentralized way. This has revived the notion and facilitated the creation of smart contracts (also called blockchain contracts, digital contracts, or self-executing contracts) that were first proposed by Nick Szabo in 1994 [1]. Smart contracts are self-executing contracts with the terms of the agreement between interested parties. The contracts are written in the form of program codes that exist across a distributed, decentralized blockchain network. Smart contracts allow transactions to be conducted between anonymous or untrusted parties without the need for a central authority [2].

Blockchain technology represented by Bitcoin and other cryptocurrencies is called blockchain 1.0, which has the typical features of decentralization, tamper-resistant, anonymity and auditability. However, writing contracts with complex logic is not possible due to the limitations of Bitcoin scripting language (Bitcoin scripting language has only 256 instructions, in which, 15 are currently disabled, and 75 are reserved). Due to limited functionality, Bitcoin can only be considered as the prototype of smart contracts. Newly emerging blockchain platforms such as Ethereum [3] embrace the idea of running user-defined programs on the blockchain, thus creating an expressive customized smart contracts with the help of Turing-complete programming language. The codes of Ethereum smart contract are written in a stack-based bytecode language and executed in Ethereum Virtual Machine (EVM). Several high-level languages such as Solidity[1] and Serpent[2] can be used to write Ethereum smart contracts. The code of those languages can then be compiled into EVM bytecodes to be run. Ethereum is currently the most popular platform for developing smart contracts, hence it is called Blockchain 2.0 [4]. In addition to Ethereum, there are some other platforms which can be utilized to develop smart contracts, such as Hyperledger Fabric [5], Corda [6] and BigchainDB [7], etc.

The correct implementation of smart contracts is enforced by the consensus protocols [8]. The contracts can encode any pre-defined rules and execute the corresponding operations when trigger conditions are satisfied. Thus, smart contracts can be applied in many fields, including intelligent assets (e.g., Slock.it[3] is a German company that utilizes Ethereum-based smart contracts for renting, selling or sharing anything without the involvement of intermediaries) and self-enforcing or autonomous governance applications (e.g., digital property management such as ujomusic[4], e-voting, and supply chain) [4].

Despite the expressiveness of smart contracts, they are facing many technical challenges. A well-known example is that in June 2016, the “Recursive calls attack” exploited the DAO (DAO[5] is an abbreviation for decentralized autonomous organization which is used as an investor-directed venture capital fund) to siphon off one third of the DAO’s funds to a subsidiary account (the affected Ether had a value of about $50M [9]). The Ethereum community had to implement the hard fork (which is a radical change to the protocol that makes previously invalid blocks/transactions valid, or vice-versa) for

[1] Solidity. https://solidity.readthedocs.io
[2] Serpent. https://github.com/ethereum/wiki/wiki/Serpent
[3] Slock.it. https://slock.it/
[4] Ujomusic. https://ujomusic.com/
[5] DAO. https://www.ethereum.org/dao

978-1-5386-4452-2/18/$31.00 ©2018 IEEE

the Ethereum blockchain to restore the affected funds. However, this was controversial because it violated the *code is law* principle. M. Alharby & A. van Moorsel [4] identified several research gaps in smart contracts research, e.g., lacking of studies on scalability and performance issues, lacking of research on tackling criminal activities, lacking of implementations to deploy and run smart contracts besides Ethereum, and so on.

The main aim of this study is to give an overview of smart contracts, including the concept and architecture, applications and future trends, etc.

The rest of this paper is organized as follows. Section II systematically introduces the smart contracts, including the basic framework, operating mechanisms, platforms and programming languages, etc. Section III presents some typical application scenarios of smart contracts. Section IV summarizes the current challenges faced by smart contracts. Section V presents the recent research progress and discusses some possible development trends. Section VI concludes the paper.

II. Smart Contract

A. Basic Framework of Blockchain

As shown in Figure 1, a typical blockchain system generally consists of six layers, namely, data layer, network layer, consensus layer, incentive layer, contract layer and application layer.

- *Data layer*. This layer includes the underlying data blocks, related encrypted messages, and timestamp, etc.

- *Network layer*. The blockchain system usually adopts the P2P protocol that is completely distributed and can tolerate single point of failures (SPoF). Blockchain network nodes have the characteristics of equality, autonomy, and distribution [10]. All the nodes are connected in a topological structure without any centralized authoritative nodes or hierarchy.

- *Consensus layer*. Consensus layer encapsulates various types of consensus protocols. This is due to the decentralized blockchain is jointly managed and maintained by multiple parties. Some of the nodes may not be credible and therefore require support for the Byzantine Fault Tolerance (BFT). Common consensus algorithms include PoW (Proof of Work), PoS (Proof of Stake), PBFT (Practical Byzantine Fault Tolerance), etc. PoW consensus process (commonly known as mining, each node is called a miner) is as follows: each node contributes their computing resources to compete solving a SHA256 mathematical puzzle (the difficulty of the puzzle could be adjusted dynamically), the winner miner broadcasts the mined block to other nodes, then other nodes confirm its validness. If the block is validated, other miners would append this new block to their own blockchains [11]. However, PoW consumes large amounts of electricity resource, result in a huge waste of energy. Thus, researchers propose some alternative consensus protocols. For example, EOS adopts DPoS (Delegated Proof of Stake) consensus protocol which leverages the power of stakeholder approval voting to resolve consensus issues in a fair and democratic way [12]. Currently, Ethereum adopts PoW + PoS hybrid consensus mechanism and Hyperledger Fabric adopts PBFT mechanism.

- *Incentive layer*. Consensus nodes in a decentralized system are self-interested, maximizing revenue is the fundamental goal of their participating in data verification and accounting. Therefore, incentive-compatible mechanisms should be designed, so that the individual rational behavior of the consensus nodes to maximize their own profits can be incentively aligned with the overall goal of guaranteeing the safety and effectiveness of the decentralized blockchain ecosystem. Taking Bitcoin as an example, the economic incentive in Bitcoin's PoW consensus mechanism consists of two parts: the newly issued Bitcoin rewards and the transaction fees. They are awarded to the node who can first solve the mathematical puzzle and record the block successfully [13].

- *Contract layer*. The contract layer encapsulates various types of script codes, algorithms, and sophisticated smart contracts, and thus is the basis for flexible programming and manipulation of blockchain systems. Most of the cryptocurrencies, including Bitcoin and Litecoin, use non-Turing-complete scripting language which means they have no flow control, namely, no loops or conditionals. Nowadays, more complex and flexible scripting languages for smart contracts have emerged, e.g., Solidity and Serpent, which enable blockchain to support a wider range of applications of finance and social systems.

- *Application layer*. The main application in the Bitcoin system is digital currency transactions. For Ethereum platform, in addition to currency transactions, it also supports Decentralized Applications (Dapp). Dapp is an application that runs on a decentralized network such as Ethereum. Hyperledger Fabric mainly aimed at enterprise-level blockchain applications, its Dapps can be built on SDKs using programming languages such as Go, Java, Python, and Node.js [14].

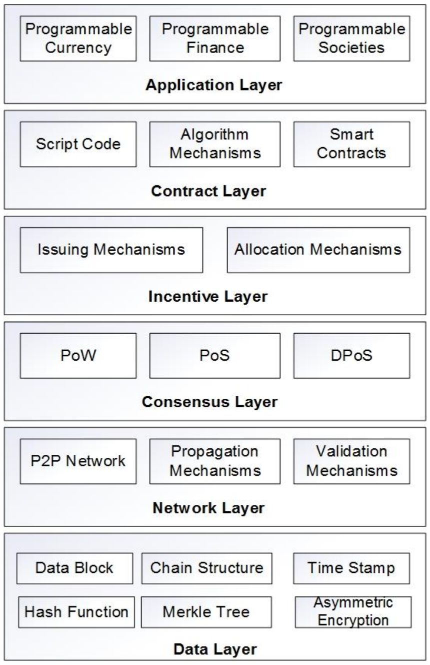

Figure 1. A basic framework of blockchain.

B. The Operating Mechanisms of Smart Contracts

Smart contracts are a set of Scenario-Response procedural rules and logic. In other words, they are decentralized, trusted shared codes that deployed on blockchain. The parties signing a contract should agree on contractual details, conditions of breach of contract, liability for breach of contract and the external verification data sources (oracles), then deploy it on the blockchain in the form of smart contract thus to automate the execution of contract on behalf of the signatories. The whole process is independent of any central agencies.

The operating mechanism of smart contracts is shown in Figure 2. Normally, after the smart contracts are signed by all parties, they are attached to the blockchain in the form of program codes (e.g., a Bitcoin transaction), and are recorded in the blockchain after being propagated by the P2P network and verified by the nodes. Smart contract encapsulates a number of pre-defined states and transition rules, scenarios that trigger contract execution (such as at a given time or a particular event occurs), responses in a particular scenario, etc. The blockchain monitors the real-time status of smart contracts and executes the contract after certain trigger conditions have been met [15].

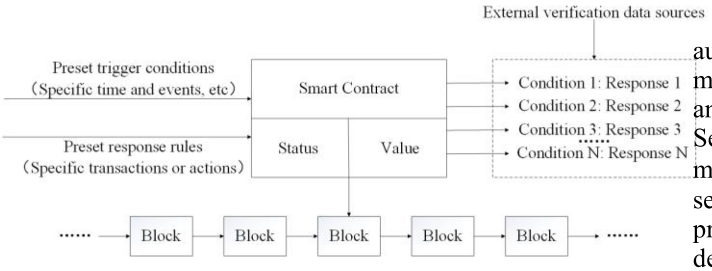

Figure 2. The operating mechanisms of smart contract.

C. Platforms and Programming Languages

Smart contracts can be developed and deployed in different blockchain platforms. Different platforms have different characteristics. In this section, we will introduce two typical platforms, namely, Ethereum and Hyperledger Fabric.

- *Ethereum*. Ethereum is a public blockchain platform on which applications run exactly as programmed without any possibility of downtime, censorship, fraud or third-party interference[6]. A Turing-complete virtual machine called Ethereum Virtual Machine (EVM) is used to execute contract bytecode (the bytecode is compiled by Solidity as mentioned before). The applications on Ethereum are run on its platform-specific cryptographic token — Ether. Ether is used in two ways, one is traded as a digital currency like Bitcoin; another is used to compensate participant nodes for the computations they performed [16].

- *Hyperledger Fabric*. Hyperledger Fabric is a blockchain framework implementation which is hosted by Linux Foundation[7]. It is a platform for distributed ledger solutions, underpinned by a modular architecture delivering high degrees of confidentiality, resiliency, flexibility and scalability. It is designed to support pluggable implementations of different components, and accommodate the complexity and intricacies that exist across the economic ecosystem. Hyperledger Fabric leverages container technology to host smart contracts called "chaincode" which comprise the application logic of the system. Besides that, "chaincode" is the only channel that interacts with the blockchain and the only source that generates the transactions.

As mentioned before, the codes of Ethereum smart contracts are written in stack-based bytecode language and executed in EVM. Several high-level languages (e.g., Solidity, Serpent and LLL) can be used to write Ethereum smart contracts. The codes will then be compiled into EVM bytecodes to be run. Hyperledger Fabric develops smart contracts using Go and Java, etc. The essence of deploying smart contracts on Hyperledger Fabric is to realize the three functions — Init, Invoke and Query in the "chaincode" interface, which are respectively used to implement contract deployment, transaction processing and transaction inquiries [4], [24].

D. Characteristics and Significance

Smart contracts have three characteristics, namely, autonomy, self-sufficiency, and decentralization. Autonomy means that after they are launched and executed, the contracts and the initiating agents need not be in further contact. Second, smart contract can be self-sufficient in their ability to marshal resources — that is, raising funds by providing services, and spending them when needed, e.g., gain processing power or storage. Third, smart contracts are decentralized as they do not subsist on a single centralized server, they are distributed and self-executed across network nodes [17].

[6] Ethereum. https://www.ethereum.org/
[7] Hyperledger Fabric. http://hyperledger-fabric.readthedocs.io/en

Smart contracts have important implications for the blockchain. On one hand, smart contracts are the activators of blockchain that laying the foundation for the programmable financial and social systems in the era of Blockchain 2.0 and 3.0. On the other hand, the automation and programmable features of smart contracts make it possible to encapsulate the complex behavior of nodes in a distributed blockchain system, which helps to promote the applications of blockchain technology in distributed artificial intelligence systems, and thus makes it possible to build various types of decentralized autonomous organization (DAO), decentralized autonomous corporation (DAC), and decentralized autonomous society (DAS) in the future.

III. APPLICATION SCENARIOS OF SMART CONTRACTS

There are various application scenarios where smart contract can be applied to. Some of these applications are as follows:

- *Financial transactions*. Smart contract is particularly suitable to business models such as equity crowdfunding, peer-to-peer lending (P2P lending) and online insurance. Traditional financial trade need to be coordinated by central agencies such as central clearing institutions or exchanges, while the agility feature of smart contract can greatly reduce transaction costs and increase efficiency, thus avoiding cumbersome clearing and delivery [18], [19].
- *Prediction markets*. Prediction markets have been proven to be able to provide better future forecasts, more direct hedging and speculation mechanisms. Due to the distributed consensus verification and immutability, smart contract can be used in prediction markets. Two typical applications are Augur and Gnosis. Augur [8] manage to create a stunningly accurate forecasting tool with the help of blockchain. Gnosis provides participants with a playful environment to try out trading in markets and win GNO tokens as a reward for successful predictions [20]. In addition, smart contract can also be used for voting or gambling.
- *Internet of Things (IoT)*. The combination of smart contracts and IoT can not only facilitate the sharing of information between devices, but also allow people to automate time-consuming workflows in a cryptographically veritable manner [21]. Slock.it proposed the first autonomous lock that people can open with token. The owners of the lock only decide on two numbers — the deposit cost and the rental cost. The user of the lock scans it and sees what he or she has to pay in terms of deposit. When the deposit is received by the lock, control is granted. The users can then open and close the lock as often as they want to. When people no longer need the lock, they get back their deposit minus the cost of rental, which is transferred to the owner of the lock [22]. Besides houses and apartments renting, the lock can be applied to cars, bikes, padlocks, etc. Smart contract can also be used on smart grid that the renewable energy could be bought and sold in a P2P market, thus reducing intermediate costs [23].

There are also some possible application scenarios of smart contracts such as digital rights management, social media platforms, cloud storage, supply chain, intelligent transportation, etc.

IV. CHALLENGES

Smart contracts have distinct characteristics compared with traditional distributed applications. On one hand, many smart contract platforms, such as Ethereum, operate on the public networks where arbitrary participants can join. One the other hand, due to the immutable nature of blockchain, contracts cannot be modified once they are deployed, so hackers can exploit this vulnerability to attack. In addition, since smart contract is generally used to transfer digital assets, security and privacy are of paramount importance [24]. In this section, we list some of the challenges that smart contract faces currently as follows:

- *Reentrancy vulnerability*. This problem occurs when an attacker utilizes a recursive call function to conduct multiple repetitive withdrawals, while their balances are only deduced once [4]. This may result in unexpected behaviors, even eventually consuming all the gas. The most notorious case is the DAO attack we talked in section I.
- *Transaction-Ordering Dependence (TOD)*. This occurs when several dependent transactions that invoke the same contract are included in one block. We know that miners can set arbitrary order between transactions, namely, contract's final state rest with how the miner sorts the transactions. Thus, an adversary can successfully launch an attack if those transitions were not executed in the right order [4], [24].
- *Timestamp Dependence*. Generally, the timestamp is set to the current time of the miner's local system. However, the miner can change this value while still having other miners accept the block. The security problem arises when the timestamp is used as a triggering condition to perform specific actions (e.g., sending money) because the attacker can use different block timestamps to manipulate the result of the contract. We call such contract as timestamp-dependent contract.
- *Lacking of trustworthy data feeds*. As mentioned in [4], smart contracts sometimes require information from external resources. However, the reliability of the information can not be guaranteed.
- *Privacy issues*. Since all transaction history is stored on the blockchain and is visible to anyone, it is theoretically possible to obtain user's private information by analyzing transaction graph structures, we call this deanonymization attack [25].

[8] Augur. https://www.augur.net/

V. Recent Advances & Future Trends

In this section, we describe the recent advances and present the future development trends of smart contracts.

A. Recent Advances

To deal with the above challenges, researchers have proposed several solutions. Natoli et al. [26] proposed using Ethereum-based functions (e.g., SendIfReceived) to enforce the order of transactions to avoid TOD. L. Luu et al. [24] proposed a symbolic execution tool called Oyente to search potential security bugs. Besides, they also proposed ways to enhance the operational semantics of Ethereum to make contracts less vulnerable. Aiming at lacking of trustworthy data feeds, F. Zhang et al. [27] proposed the Town Crier solution which provides credible data for smart contracts from trusted web servers. A. Kosba et al. [28] proposed a blockchain model called Hawk that does not store financial transactions clearly on the blockchain, thus retaining users' transaction privacy.

B. Future Trends: Parallel Blockchain and Smart Contracts

The rapid development of the Internet in recent years and its deep coupling with the physical world have fundamentally changed the mode of production, people's lifestyle, and the management style of modern society. The development trend of the future society is bound to a transformation from Cyber-physical systems (CPS) to Cyber-physical-social systems (CPSS) that social and human factors must be taken into account. At present, the parallel society based on CPSS has already begun to emerge, its core and essential characteristics are virtual-real interaction and feedback regulation [29], [30].

Blockchain and its smart contracts are the basic infrastructure for realizing CPSS-based parallel society. Its main contribution is to provide a set of effective decentralized data structures, interactive mechanisms and computing models for distributed social systems and distributed artificial intelligence, thus laying a solid foundation for future parallel society [31], [32]. For example, modern centralized social systems inevitably have the characteristic of "Merton's systems" due to their high degree of engineering complexity and social complexity, namely, uncertainty, diversity and complexity (UDC), so central agencies and policy makers may gain illegally profits by their privilege. Fortunately, smart contracts can help realize the software-defined social systems. The basic idea is to remove the centralized authorities, deploy the unpredictable behavior on the blockchain in the form of contractual codes, thus to guarantee they are executed automatically and hard to forge or tamper with. Consequently, to some extent, the "Merton's system" can be transformed into the "Newton's system" which could be comprehensively observed, actively controlled, and accurately predicted [33].

The ACP approach (Artificial societies + Computational experiments + Parallel execution) is by far the only systematic and complete research framework in the field of parallel management [34], [35], [36]. The ACP approach can be naturally combined with smart contracts, thus to realize the parallel management of socioeconomic systems [37], [38]. Y. Yuan & F. Y. Wang [39] proposed the conceptual framework, fundamental theory and research methodology of parallel blockchain: Firstly, consensus nodes and smart contracts in the blockchain will form DAC and DAO by participating in various types of Dapp, and eventually form DAS (DAS corresponds to the artificial society in ACP); Secondly, the programmable feature of smart contracts allows the blockchain to perform various "WHAT-IF" types of experimental design, scenario deduction and results evaluation, through which the optimal decision can be obtained either automatically or semi-automatically; Lastly, through the parallel interaction and co-evolution of the actual systems and artificial systems, the collaborative optimization of social management and decision-making which combines description, prediction and prescriptiveness will come true. In short, the ultimate goal of parallel blockchain is to realize the knowledge automation of the blockchain ecosystems [40].

VI. Conclusion

With the rapid development of blockchain technologies, the emerging smart contracts have become a hot research topic in both academia and industry. The immutable and irreversible characteristics of smart contracts can help people exchange money, shares, intellectual property, etc. in a transparent, conflict-free way while avoiding the interference of third-party. Thus, smart contracts will get widely used in financial and social systems in the near future. In this paper, we present an overview on smart contract, including its concept, architecture, and application scenarios. We also discuss challenges that the smart contract faces, and present its future trends. We plan to conduct further investigations on parallel blockchain and the related smart contract applications in the future.

Acknowledgment

This work was supported by National Natural Science Foundation of China (71472174, 61533019, 71232006, 61233001, 61702519, 71702182), Qingdao Think-Tank Foundation on Intelligent Industries.

References

[1] N. Szabo, "Smart Contracts: Building Blocks for Digital Markets," 1996. [Online]. Available: http://www.fon.hum.uva.nl

[2] D. Tapscott, and A. Tapscott, Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World. London: Portfolio, 2016.

[3] V. Buterin, "A next-generation smart contract and decentralized application platform," 2017. [Online]. Available: http://github.com/ethereum/wiki/wiki/White-Paper/

[4] M. Alharby, and A. van Moorsel, "Blockchain-based Smart Contracts: A Systematic Mapping Study," *arXiv:1710.06372*, 2017.

[5]" Hyperledger Fabric project," 2017. [Online]. Available: https://www.hyperledger.org/projects/fabric

[6]" Corda: Frictionless Commerce," 2017. [Online]. Available: https://www.corda.net/

[7]"BigchainDB: The scalable blockchain database powering IPDB," 2017. [Online]. Available: https://www.bigchaindb.com/

[8] N. Szabo, "The idea of smart contracts," 1997. [Online]. Available: http://szabo.best.vwh.net/smart_contracts_idea.html

[9] N. Popper, "A Hacking of More Than $50 Million Dashes Hopes in the World of Virtual Currency," 2016. [Online]. Available: https://www.nytimes.com/2016/06/18/business/dealbook/hacker-may-have-removed-more-than-50-million-from-experimental-cybercurrency-project.html

[10] A. Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies. Sebastopol, CA: O'Reilly Media, 2014.

[11] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2009. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[12] "EOS.IO Technical White Paper v2," 2018. [Online]. Available: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md

[13] J. Fan, L. T. Yi, and J. W. Shu, "Research on the technologies of Byzantine system," *Journal of Software*,vol. 24, no. 6, pp. 1346-1360, 2013.

[14] Q. F. Shao, C. Q. Jin, Z. Zhang, and W. N. Qian, "Blockchain: Architecture and Research Progress," *Chinese Journal of Computer*,vol. 40, no. 157, pp. 1-21, 2017.

[15] K. Delmolino, M. Arnett, A. Kosba, A. Miller, and E. Shi, "Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab," in *International Conference on Financial Cryptography and Data Security*, Springer, Berlin, Heidelberg, 2016, vol. 9604, pp. 79-94.

[16] J. Chow, "Ethereum, Gas, Fuel, & Fees," 2017. [Online]. Available: https://media.consensys.net/ethereum-gas-fuel-and-fees-3333e17fe1dc

[17] M. Swan, Blockchain: Blueprint for a New Economy. Sebastopol, CA: O'Reilly Media, 2015.

[18] I. Allison, "Game-changers FreeMyVunk and Digix allow video gamers to trade virtual assets for physical gold," 2015. [Online]. Available: http://www.ibtimes.co.uk/game-changers-freemyvunk-digix-allow-video-gamers-trade-virtual-assets-physical-gold-1534436

[19] A. Mizrahi, "Everex helps migrant workers send remittance back to Myanmar via Ethereum," 2016. [Online]. Available: https://www.financemagnates.com/cryptocurrency/education-center-2/everex-helps-migrant-workers-send-remittance-back-myanmar-via-ethereum/

[20] A. Hertig, "Ethereum Prediction Market Service Takes First Steps With Beta Launch," 2016. [Online]. https://www.coindesk.com/ethereum-prediction-market-service-takes-first-steps-beta-launch/

[21] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things, " in *IEEE Access*, vol. 4, pp. 2292-2303, 2016.

[22] I. Allison, "Ethereum-based Slock.it reveals first ever lock opened with money," 2015. [Online]. Available: http://www.ibtimes.co.uk/ethereum-based-slock-reveals-first-ever-lock-opened-money-1527014

[23] A. Rutkin, "Blockchain-based microgrid gives power to consumers in New York," 2016. [Online]. Available: https://www.newscientist.com/article/2079334-blockchain-based-microgrid-gives-power-to-consumers-in-new-york/

[24] L. Luu, D. H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making Smart Contracts Smarter," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*, Vienna, Austria, 24 - 28 Oct., 2016, pp. 254-269.

[25] D. Ron, and A. Shamir, "Quantitative Analysis of the Full Bitcoin Transaction Graph," in *Sadeghi AR. (eds) Financial Cryptography and Data Security (FC 2013)*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol. 7859, pp. 6-24.

[26] C. Natoli, and V. Gramoli, "The Blockchain Anomaly," in *IEEE 15th International Symposium on Network Computing and Applications (NCA)*, Cambridge, MA, USA, 31 Oct. - 2 Nov., 2016, pp. 310-317.

[27] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, "Town Crier: An Authenticated Data Feed for Smart Contracts," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*, Vienna, Austria, 24 - 28 Oct., 2016, pp. 270-282.

[28] A. Kosba, A. Miller, E. Shi, Z. K. Wen, and C. Papamanthou, "Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts," in *2016 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, 22 May - 26 May, 2016, pp. 839-858.

[29] X. Wang, L. X. Li, Y. Yuan, P. J. Ye, and F. Y. Wang, "ACP-based social computing and parallel intelligence: Societies 5.0 and beyond," *CAAI Transactions on Intelligence Technology*,vol. 1, no. 4, pp. 377-393, 2016.

[30] F. Y. Wang, J. J. Zhang, X. H. Zheng, X. Wang, Y. Yuan, X. X. Dai, J. Zhang and L. Q. Yang, "Where does AlphaGo go: from church-Turing thesis to AlphaGo thesis and beyond," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 2, pp. 113-120, 2016.

[31] Y. Yuan, and F. Y. Wang, "Blockchain: the state of the art and future trends," *Acta Automatica Sinica*, vol. 42, no. 4, pp. 481-494, 2016.

[32] M. Z. Kang, and F. Y. Wang, "From parallel plants to smart plants: intelligent control and management for plant growth," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 2, pp. 161-166, 2017.

[33] F. Y. Wang, "Software-defined systems and knowledge automation: a parallel paradigm shift from Newton to Merton," *Acta Automatica Sinica*, vol. 41, no. 1, pp. 1-8, 2015.

[34] F. Y. Wang, "Artificial societies, computational experiments, and parallel systems: a discussion on computational theory of complex social-economic systems," *Complex Systems and Complexity Science*, vol. 1, no. 4, pp. 25-35, 2004.

[35] D. Wen, Y. Yuan, and X. R. Li, "Artificial societies, computational experiments, and parallel systems: an investigation on a computational theory for complex socioeconomic systems," *IEEE Transactions on Services Computing*, vol. 6, no. 2, pp. 177-185, 2013.

[36] S. Wang, X. Wang, P. J. Ye, Y. Yuan, S. Liu, and F. Y. Wang, "Parallel crime scene analysis based on ACP approach," *IEEE Transactions on Computational Social Systems*, to be published.

[37] Y. Yuan, and F. Y. Wang, "Towards blockchain-based intelligent transportation systems," in *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, 2016, pp. 2663-2668.

[38] F. Y. Wang, D. J. Zeng, and Y. Yuan, "An ACP-based approach for complex analysis of E-commerce system," *Complex Systems and Complexity Science*, vol. 5, no. 3, pp. 1-8, 2008.

[39] Y. Yuan, and F. Y. Wang, "Parallel blockchain: concept, methods and connotation analysis," *Acta Automatica Sinica*, vol. 43, no. 10, pp. 1703-1712, 2017.

[40] S. Zeng, S. Wang, Y. Yuan, X. C. Ni, and Y. J. Ouyang, "Towards knowledge automation: a survey on question answering systems," *Acta Automatica Sinica*, vol. 43, no. 9, pp. 1491-1508, 2017.