Blockchain-Enabled Smart Contracts: Architecture, Applications and Future Trends

Shuai Wang[1,2], Liwei Ouyang[1,2], Yong Yuan*[1,3] (*Corresponding author, Senior Member, IEEE*), Xiaochun Ni[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, ouyangliwei2018, yong.yuan, xiaochun.ni, feiyue.wang}@ia.ac.cn

***Abstract*—In recent years, the rapid development of cryptocurrencies and their underlying blockchain technology has revived Nick Szabo's original idea of smart contracts, i.e., computer protocols that are designed to automatically facilitate, verify, and enforce the negotiation and implementation of digital contracts without central authorities. Smart contracts can find a wide spectrum of potential application scenarios in the digital economy and intelligent industries including financial services, enterprise management, healthcare, Internet of Things (IoT) and supply chain, among others, and also have been integrated into the main-stream blockchain-based development platforms, such as Ethereum and Hyperledger. However, smart contracts are still far from mature, and major technical challenges such as security issues and privacy disclosure are still awaiting further research efforts. For instance, the most widely-known case might be "The DAO Attack" in June 2016, which led to more than 50 million US dollars Ether transferred illegally into an adversary's account. In this paper, we strive to present a systematic and comprehensive overview of blockchain-enabled smart contracts, aiming at stimulating further research towards this emerging research area. We first introduced the operating mechanisms and main-stream platforms of blockchain-enabled smart contracts, and proposed a research framework for smart contracts based on a novel six-layer architecture. Second, we presented some typical application scenarios for smart contracts. Third, both the technical and legal challenges, as well as the recent research advances, are listed. Towards the end, we discussed the future development trends of smart contracts. This paper is aimed at providing helpful guidance and reference for future research efforts.**

Keywords—blockchain; smart contracts; six-layer architecture; parallel blockchain

I. INTRODUCTION

The term "smart contract" was first coined in mid-1990s by computer scientist and cryptographer Nick Szabo, who defined a smart contract as "a set of promises, specified in digital form, including protocols within which the parties perform on these promises [1]." In his famous example, Szabo analogized smart contracts to vending machines: machines take in coins, and via a simple mechanism (e.g., finite automata), dispense change and product according to the displayed price. Smart contracts go beyond the vending machine by proposing to embed contracts in all sorts of properties by digital means [2]. Szabo also expected that through clear logic, verification and enforcement of cryptographic protocols, smart contracts could be far more functional than their inanimate paper-based ancestors. However, the idea of smart contract did not see the light till the emergence of blockchain technology, in which the public and append-only distributed ledger technology (DLT) and the decentralized consensus mechanism make it possible to implement smart contracts in its true sense.

Generally speaking, smart contracts can be defined as the computer protocols that digitally facilitate, verify and enforce the contracts made between two or more parties on blockchains. Once they are created, they act autonomously. For this reason, many prefer to use the term *smart agents* or *software agents* to characterize smart contracts.

As smart contracts are typically deployed on and secured by blockchain, they have some unique characteristics. First, the program code of a smart contract will be recorded and verified on blockchain, thus making the contract immutable and tamper resistant. Second, the execution of a smart contract is enforced automatically among anonymous, trustless individuals without centralized control and coordination of third-party authorities. Third, a smart contract, like an intelligent agent, might have its own cryptocurrencies or other digital assets, and transfer them when pre-defined conditions are triggered [3].

It's worth noting that Bitcoin is widely recognized as the first cryptocurrency that can support basic smart contracts, in the sense that its transactions will be validated only if certain conditions are satisfied. However, designing "smart" contracts with complex logic is not possible due to the limitations in the Bitcoin scripting language that only features some basic arithmetic, logical, and crypto operations.

Ethereum[1] is a public blockchain platform that supports advanced and customized smart contracts with the help of Turing-complete virtual machine called Ethereum Virtual Machine (EVM). EVM is the runtime environment for smart contracts, and every node in the Ethereum network runs an EVM implementation and executes the same instructions. Several high-level programming languages, such as Solidity[2], Serpent[3],

[1] Ethereum. https://www.ethereum.org/
[2] Solidity. http://solidity.readthedocs.io/en/latest/

and LLL[4], can be used to design Ethereum smart contracts, and then the contract code is compiled down to EVM bytecode and deployed onto the blockchain for execution. Ethereum is currently the most popular development platform for smart contracts, and can be used to design various kinds of decentralized applications (DApps), e.g., digital rights management, crowdfunding, gambling, and so on.

Although smart contracts have made great progresses in recent years, they are still faced with many challenges. A well-known event is that in June 2016, The DAO, a decentralized investor-directed venture capital fund secured by blockchain and smart contracts, was attacked by exploiting a severe bug through the "Recursive call attack". The attacker drained more than US$50 million Ether into a "child DAO" that has the same structure as The DAO. At last, a hard fork of the Ethereum software was forced to be implemented to claw back the funds from the attacker. However, this hard fork was controversial because it is widely believed to violate the *code is law* principle in the spirit of blockchain technology. In addition to security vulnerabilities, other challenges include lacking trustworthy data feeds, privacy leakage and legal issues, etc.

The main aim of this paper is to offer a comprehensive overview of smart contracts research, including the operating mechanisms, basic framework, application fields, the existing problems and future trends.

The rest of this paper is organized as follows. Section II systematically introduces the smart contracts, including the operating mechanisms, main-stream development platforms, and a basic research framework of smart contracts which employs a six-layer architecture is proposed. Section III presents some typical application fields of smart contracts, e.g., financial services, management, healthcare, IoT, etc. Section IV summarizes the current open challenges faced by smart contracts and also the recent research progress. Section V discusses some possible development trends. Section VI concludes the paper.

II. SMART CONTRACTS

We will give an overview of smart contracts in this section. First, we make a brief introduction to blockchain, and then present the operational mechanism of smart contracts based on two main-stream platforms — Ethereum and Hyperledger Fabric. We also propose a basic research framework of smart contracts.

A. A Brief Introduction to Blockchain

Blockchain was conceptualized by Satoshi Nakamoto, who proposed a solution to the double-spending problem using a peer-to-peer network (known as Bitcoin) in 2008. Blockchain is a continuously growing list of records, called blocks, which are linked and secured using cryptography. Blockchain adopts the P2P protocol that can tolerate single point of failure in the network communications protocol. The consensus mechanism ensures a common, unambiguous ordering of transactions and blocks, and guarantees the integrity and consistency of the blockchain across geographically distributed nodes. By design, blockchain has characteristics of decentralization, tamper-resistant, and auditability [4]. According to different level of access permission, blockchains can be divided into three types: public blockchains (such as Bitcoin and Ethereum), consortium blockchains (such as Hyperledger and Corda[5]), and private blockchains. Blockchain has a wide spectrum of application scenarios, including trade finance, identity management, etc.

Smart contracts are software programs that self-execute complex instructions on blockchains. Once a smart-contract has been signed and deployed onto blockchain, it is essentially immutable, and cannot be manipulated by those involved in the agreement. In the next section, we will discuss the operational mechanism of smart contracts.

B. The Operational Mechanism of Smart Contracts

The operational mechanism of smart contracts is shown in Fig. 1. Smart contracts generally have two attributes: value and state. The triggering scenarios and response processing rules for the contract terms are preset using "IF-THEN" statements. Smart contracts are agreed upon and signed by all stakeholders and submitted to blockchain network, then they are propagated via the P2P network, verified by the miners and stored in a specific block of the blockchain. The creators of the contracts get the returned contract addresses and interfaces, then users can invoke a contract by initiating a transaction. Miners are motivated by the system's incentive mechanism and will contribute their computing resources to verify the transactions. More specially, after the miners download the required data from the blockchain, they execute the contract code in the local virtual machine. By querying the trusted external data source (a.k.a., oracles) and checking the system state, the miners determine whether the current scenario satisfies the contract trigger condition. If the trigger condition is met, the contract code is strictly executed and the system state is updated accordingly. After that, the miners pack the transactions and create new block. The new block is validated by the consensus algorithm and appended to the existing blockchain.

[3] Serpent. https://github.com/ethereum/wiki/wiki/Serpent

[4] LLL. http://lll-docs.readthedocs.io/en/latest/lll_introduction.html

[5] Corda. https://docs.corda.net/

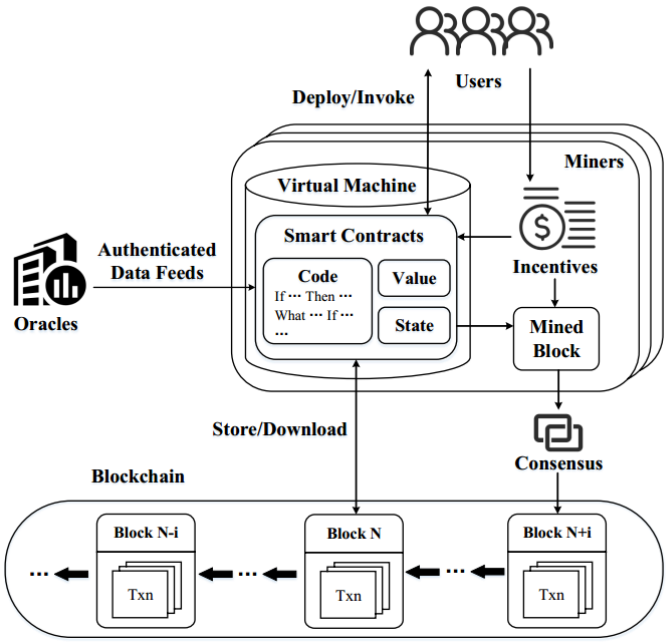

Fig. 1. The operational mechanism of smart contracts.

Next, we take Ethereum and Hyperledger Fabric platforms as examples to introduce the operational process of smart contracts.

1) Ethereum. Ethereum is currently the most widely used smart contract development platform that can be viewed as a transaction-based state machine: it begins with a genesis state and incrementally executes transactions to morph it into some final state. It is this final state which we accept as the canonical "version" in the world of Ethereum [5]. Unlike the UTXO model of Bitcoin, Ethereum introduced the concept of accounts. There are two types of accounts: externally owned accounts (EOAs) and contract accounts. The difference is that the former is controlled by private keys without code associated with them, while the latter is controlled by their contract code with associated code.

Users can only initiate a transaction through an externally owned account. The transaction can include binary data (its payload) and Ether. If the recipient of a transaction is the zero-account $\varnothing$, a smart contract is created. Or if the recipient is a contract account, the account will be activated and its associated code is executed on the local EVM (the payload is provided as input data). The transaction is then broadcast to the blockchain network where miners will verify it [6], as shown in Fig.2.

In order to avoid issues of network abuse and to sidestep the inevitable questions stemming from Turing completeness, all programmable computation (e.g., creating contracts, making message calls, utilizing and accessing account storage and executing operations on the virtual machine) in Ethereum is subject to fees — a reward for miners who contribute their computing resources. The unit used to measure the fees required for the computation is called gas [5].

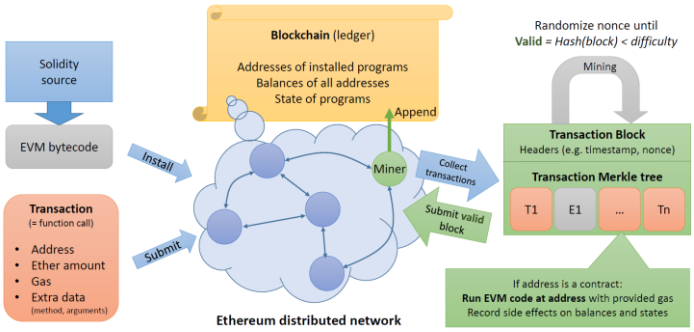

Fig. 2. Overview of workflow in the Ethereum network [6].

2) Hyperledger Fabric. Hyperledger Fabric [6] is a blockchain framework implementation and one of the Hyperledger projects hosted by The Linux Foundation. Rather than the permissionless blockchian such as Bitcoin and Ethereum that allows unknown identities to participate in the network, Hyperledger Fabric is permissioned because only a collection of business-related organizations can join in through a trusted Membership Service Provider (MSP), and its network is built up from the peers that are owned and contributed by these organizations. Peers are hosts for blockchain ledgers and chaincodes (smart contracts). The ledger is the sequenced, tamper-resistant record of all state transitions in the Fabric. State transition is a result of chaincode invocation (transaction) submitted by participating parties. Each transaction results in a set of asset key-value pairs that are committed to the ledger as creates, updates, or deletes. Chaincode enforces the rules for reading or altering key-value pairs or other state database information. As shown in Fig.3, The transaction workflow of Hyperledger Fabric consists of three phases as follows:

Proposal. An application sends a transaction proposal to different organizations' endorsing peers (also called endorsers who validate transactions against endorsement policies and enforce the policies). The proposal is a request to invoke a chaincode function so that data can be read and/or written to the ledger. The transaction results include a response value, read set, and write set. The set of these values, along with the endorsers' signatures are returned to the application as a transaction proposal response.

Packaging. The application verifies the endorsers' signatures and check if the proposal responses are the same. Then, the application submits the transaction to Ordering Service (orderer) to update the ledger. The orderer sorts the transactions it received from the network, and packages batches of transactions into block ready for distribution back to all peers connected to it.

Validation. The peers connected to the orderer validate every transaction within the block to ensure that it has been consist-

[6] Hyperledger Fabric. https://www.hyperledger.org/projects/fabric

ently endorsed by required organizations according to the endorsement policy. It's worth noting that this phase does not require the running of chaincode — this is only done in proposal phase. After validation, each peer appends the block to the chain, and the ledger is updated.

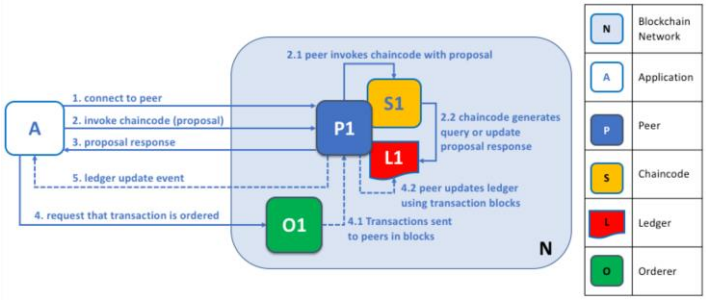

Fig. 3. The transaction workflow of Hyperledger Fabric [7]

C. A Basic Research Framework of Smart Contracts

On the basis of the operational mechanism of the mainstream smart contract platforms, in this paper, we propose a basic research framework for smart contracts. As shown in Fig. 4, the framework employs a six-layer architecture, namely, infrastructures layer, contracts layer, operations layer, intelligence layer, presentations layer and applications layer. The details are as follows:

- *Infrastructures layer*. The infrastructures layer encapsulates all the infrastructures needed to implement smart contracts and their applications. Smart contracts can be narrowly defined as computer programs running on the distributed ledgers, their execution and interaction rely on some key components such as consensus algorithms, incentive mechanisms and communication networks. In the process of contracts development, deployment, and invocation, a variety of development tools are involved, including programming languages, integrated development environments (IDE), development frameworks, clients, wallets, and interactive tools, etc. At the same time, in order to guarantee the security of blockchain network, smart contracts are generally executed in fully isolated virtual machines (such as EVM in Ethereum and Docker container in Hyperledger Fabric). In addition, oracles can serve as external data feeds that provide smart contracts with information about the real-world occurrences.

- *Contracts layer*. The contracts layer can be regarded as a static database of smart contracts as it encapsulates all the rules about contracts execution, invocation and communication. When a smart contract is being designed, at first all the contractors shall negotiate and determine the contract's contents which may include legal provisions, business logics and intention agreements. Then, the programmers will translate the contract contents described in natural language into program code, e.g., a serious of "IF-THEN"-typed scenario-response rules. Moreover, interaction criteria should also be enacted in this layer which can specify the guidelines for interactions between contracts and contracts (or contracts and users).

- *Operations layer*. The operations layer encapsulates all the dynamic operations on the static contracts, which is the key to the correct, safe and efficient operation of smart contracts. From the perspective of smart contracts life cycle, mechanism design applies information and incentive theory to enable the contracts to be implemented efficiently; formal verification and security inspection prove the correctness of the contracts code through strict mathematical proof before the contracts are deployed onto the blockchains, and ensure that the code will be executed according to the programmers' intentions; maintenance and updates guarantee the normal running of smart contracts and upgrade them when necessary. At the end of the smart contracts' life cycle or when a high-risk vulnerability occurs, self-destruction is conducted to insure network security.

- *Intelligence layer*. As mentioned earlier, smart contracts running on the blockchain network can be seen as software agents that act on behalf of their users. With the development of Artificial Intelligence (AI) technology, agents will have a certain degree of intelligence, such as perception, reasoning, and learning by virtue of cognitive computing, reinforcement learning, etc. Hence, those agents are not only autonomous as they have capabilities of tasks selection, prioritization, and goal-directed behaviors (sometimes referred to as Beliefs-Desires-Intentions, or BDI), but also have sociability through communication, cooperation and negotiation with each other. The learning and collaboration results will also be fed back to the contracts layer and operations layer, thus to optimize the contract designs and operation schemes, and finally realizing the autonomous multi-agent systems. In short, future smart contracts should have "WHAT-IF"-typed deduction, computational experiments, and decision-making capabilities in unknown scenarios.

- *Presentations layer*. The presentations layer encapsulates all the manifestations of smart contracts. As mentioned before, the automatization and programmability of smart contracts makes it possible to encapsulate all the complex behaviors of network nodes into a distributed system, so smart contracts can be regard as the application interfaces that enable blockchains to embed different application scenarios. A typical case is the emerging DApps built on Ethereum. Furthermore, the multi-agent systems in intelligence layer will continue to evolve to form various decentralized autonomous organizations (DAOs) (sometimes labeled as decentralized autonomous corporations, DACs). DAOs are organizations that are powered and run by smart contracts, and their data such as financial records and regulations are all stored on blockchains. DAOs can reduce transaction costs and introduce the possibility of finding new ways of aligning interests and governing groups of people in a more decentralized manner. Thus, DAOs are

[7] Hyperledger Fabric Docs. http://hyperledger-fabric.readthedocs.io/en/release-1.1/peers/peers.html

expected to bring disruptive influence to the traditional management paradigms which are typically in a top-down hierarchical structures [7]. The supreme presentation of DAOs are called decentralized autonomous societies (DAS), which are the foundation for programmable societies.

- *Applications layer*. Built upon the DAOs or other forms of presentations of smart contracts, applications for various industries can be established, such as financial services, IoT, healthcare, supply chain, etc. We will describe in detail in section III.

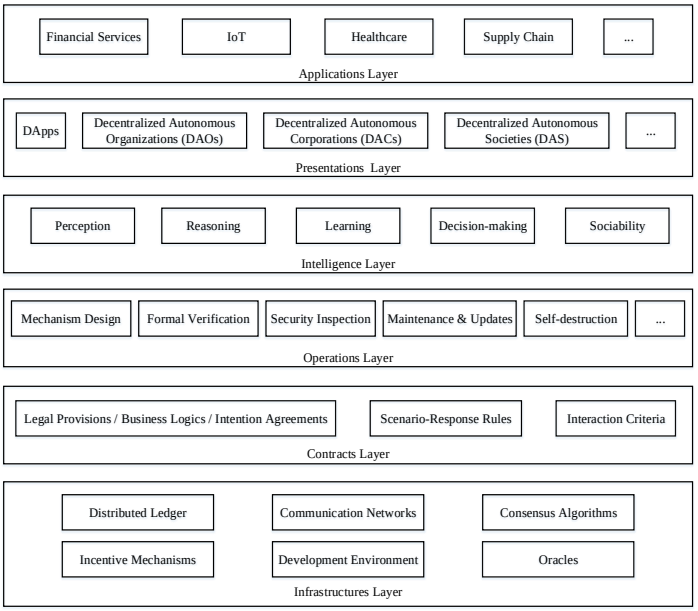

Fig. 4. An architecture of smart contracts.

III. Application Fields of Smart Contracts

Currently, smart contracts applications are springing up. This section will take financial services, management, healthcare, Internet of Things (IoT) and supply chain as examples to introduce the application fields of smart contracts.

A. Financial Services

For financial institutions, building trust between customers and businesses in digital transactions is paramount to continued loyalty and growth. Blockchain and smart contracts enable increased visibility and trust across the participants while bring huge savings in infrastructures, transactions, and administrative costs. The following are some typical applications of smart contracts in financial services.

- *Clearing and settlement*. Nowadays, banking is hampered by the inefficiencies of traditional processes of clearing and settlement for financial assets [8-9]. Major markets in US, Canada and Japan still have a 3-day settlement cycle (T+3) [10] and the transaction execution involves many institutions such as securities depositories and collateral management agencies. The centralized clearing entails labor-intensive activities that include various approvals and/or complex internal and external reconciliations. Blockchain enables bilateral peer-to-peer execution of clearing business logic using smart contracts. Currently, over 200 banks, financial institutions, regulators and trade associations within the R3 consortium participate in the testing of clearing and settlement process on Corda — R3's blockchain platform. The Australian Securities Exchange is also working on a smart contracts-based post-trade platform to replace its equity settlement system.

- *Securities and insurances*. Blockchain-powered tokenized securities can be traded in a peer-to-peer fashion without the need of an intermediary while complying with security laws, and its custody is enforced by cryptography rather than central depositories. All market participants are able to issue and transfer equity and debt seamlessly while smart contract ensures transaction security, efficiency, and transparency [11]. In insurance industry, blockchain and smart contracts could be exploited to increase the speed of claim processing as well as to reduce costs associated with the manual processing of claims. For example, a travel insurance smart contracts will automatically compensate for travelers if their flight is delayed [12-13].

- *E-commerce*. Smart contract reduces transaction costs associated with contracting, so E-commerce companies can use it for payments which takes away the need for using costly payment processors and merchant services. Merchants can also use smart contract to automate fulfillment of orders, especially for the delivery of digital goods [14]. A decentralized marketplace called ECoinmerce[8] enables photos, videos, reviews, and all other digital assets owned by the retailer or user who made it. With the help of smart contracts, any users can build, buy, sell, and sublet their digital assets on ECoinmerce.

Other applications of smart contracts in financial include property transactions, mortgage loan [15], crowd funding, prediction markets [16], etc.

B. Management

Smart contracts are expected to reduce the management discretion problems because they can provide appropriate and transparent accountability in terms of roles, responsibilities, and decision processes within/across the organizations. Here are some cases.

- *Registries and identity management*. Storing cryptographic certification of registry entries or identities on a blockchain can facilitate the access and validation. A decentralized application called uPort[9] allows users to register their own identity on Ethereum, send and request credentials, sign transactions, and securely manage keys & data. J. L. de la Rosa et al. proposed to use smart contracts to certify the proof of existence and au-

[8] ECoinmerce. https://www.ecoinmerce.io/
[9] uPort. https://www.uport.me/

thorship of intellectual properties [17]. Smart contracts can also be applied in digital rights management. For example, a DApp called Ujo Music[10] can enforce the royalty payments for a musician once his/her work is used for commercial purposes.

- *Business Process Management (BPM)*. The BPM refers to the design, execution, monitoring, and improvement of business processes for interorganizational processes. With blockchain, a vast majority of the control flow and business logic of interorganizational business processes can be compiled from process models into smart contracts that ensure the joint processes are correctly executed [18-20]. This will bring a fundamental shift toward a distributed, trustworthy BPM-related programs, projects, and operations management [21].

- *E-voting*. The current voting system is costly and inefficient. What makes it worse is that the voting results might be maliciously manipulated. Issuing smart contracts to registered citizens could lay the groundwork to provide a reliable and virtually fraud-free voting system. Beyond that, smart contracts-based E-voting enables staffs, citizens, or members to participate in collaborative decision-making processes for organizational or enterprise management with greater autonomy, decentralization and convenience. P. McCorry et al. present a smart contract implementation for the Open Vote Network (a self-tallying voting protocol) that runs on Ethereum network [22]. DApps such as Horizon State[11] and Ropsten[12] also support E-voting.

C. Healthcare

The traditional medical industry is highly dependent on paperwork, e.g., medical history, prescribed medications, diagnostic examinations, and laboratory tests. Besides, patients do not have control over the access privileges to their own medical records. Blockchain-enabled smart contracts can effectively solve these problems. On one hand, blockchain supports the encryption storage of all medical data; On the other hand, smart contracts endow patients with rights to management and control their health data. Nowadays, there are three typical applications of smart contracts in the medical field. The first is Fast Healthcare Interoperability Resources (FHIR). For example, MeDShare [23] is a blockchain-powered system that provides data provenance, auditing, and control for shared medical data in cloud repositories, its design employs smart contracts and an access control mechanism to effectively track the behavior of the data. MedRec [24] is another decentralized electronic medical records management system that facilitate interoperability among patients, public health authorities and medical researchers. The second type of application is user-oriented medical research. T.-T. Kuo et al. proposed a framework called ModelChain that can be used for healthcare predictive modeling, each participant will contribute to the model parameter estimation without revealing any private health information [25]. The last application is counterfeit drug prevention and detection. The MediLedger Project[13] aims to create a permissioned blockchain network that improves the track-and-trace capabilities for prescription drugs. Similarly, BlockMedx[14] is developing a secure e-prescribing platform for physicians to prescribe, pharmacists to fill, and patients to manage prescriptions.

D. Internet of Things (IoT) and Supply Chain

The combination of smart contracts and Internet of Things (IoT) can facilitate the sharing of resources between different types of devices, thus leading to the creation of a marketplace for device services [26]. A. Dorri et al. developed a blockchain-based smart home tier and discussed the various transactions and procedures associated with it. Simulation results indicate that the overheads introduced by the proposed method is relatively low [27-28]. Y. Zhang et al. proposed an IoT E-business model that realizes the transactions of smart properties and paid data on the IoT with the help of P2P transactions based on the smart contracts [29]. F. Knirsch presented a protocol for dynamic tariff decisions for electric vehicle charging, different charging stations can send bids for tariffs based on the pricing and the distance to the electric vehicles [30].

Supply chain is another promising area for the application of smart contracts because the traditional supply chain contracts are complex, dynamic, multi-party arrangements (there are many stakeholders in a supply chain such as producers, processing facilities, wholesalers, retailers, and consumers), with regulatory and logistical constraints [31-32]. Smart contracts and smart properties can enact payments with improved logistics visibility when linked to key supply chain events. A representative application is a cotton supply chain on which 88 bales of cotton were shipped from US to China that illustrates how smart contracts work in practice [33]. Other supply chain applications include RFID-driven contract bids and execution, meat or drugs traceability, etc.

There are some other application fields of smart contracts, e.g., distributed file storage, social media platforms, lottery and gaming [34], intelligent transportation system [35], etc.

IV. Challenges & Recent Advances

For smart contracts, security and privacy are of paramount importance. However, smart contracts currently face many challenges, e.g., security bugs, privacy disclosure, and legal issues. In this section, we list some of the typical challenges and the recent advances.

A. Challenges

1) Immutable bugs and lacking trustworthy data feeds. Due to the immutable and irreversible nature of the blockchain [36], once the smart contracts are deployed, they can no longer be altered. In other words, if there exist a bug in a smart contract, there is no direct and easy way to fix it. Besides, the operation of smart contracts requires access to the external data about real-world states and events, and authenticated data

[10] Ujo Music. https://ujomusic.com/
[11] Horizon State. https://horizonstate.com/
[12] Ropsten. https://voting-dapp-ropsten.herokuapp.com/
[13] The MediLedger Project. https://www.mediledger.com/
[14] BlockMedx. https://blockmedx.com/

feeds (also called oracles) are thus of great importance. Currently, lacking a substantive ecosystem of trustworthy data feeds is often regarded as a critical obstacle to the evolution of smart contracts [37-38].

2) Criminal Smart Contracts (CSCs). Smart contracts enable criminals to implement some illegal actions such as ransomware, underground markets, and money laundering by creating a contract executing autonomously and automatically without further interactions, which is difficult for supervision. A. Juels et al. listed three typical CSCs as follows [39].

- *Leakage/sale of secret documents*. A typical case is Darkleaks[15] — a cryptographic information black market that provides a decentralized way to leak and get paid for the secret documents or information such as proprietary software, corporate documents, and firmware drivers, etc. People bid on those secret documents, and then redeeming funds of leaker automatically releases the decryption key to unlock the whole documents.
- *Theft of cryptographic keys*. The key-theft contract rewards a perpetrator for delivery of users' private keys such as certificate authority's signing keys or the SSL/TLS certificate private keys that will bring huge economic losses to users.
- *Utilizing contracts to facilitate physical-world crimes (terrorism, arson, etc.)*. Consider the following scenario: a smart contract states that if a person can kill a politician before a certain time limit, he/she will receive a bonus. As mentioned earlier, such behavior is difficult to regulate.

3) Security bugs. Many security bugs stem from a semantic gap between the contract programmers about the underlying execution semantics and the actual semantics of the smart contracts [40]. Here are some typical bugs:

- *Transaction-Ordering Dependence (TOD).* Each block contains several transactions, and the order in which transactions are executed depends entirely on the miner. TOD occurs when several dependent transactions invoke the same contract that the miner can manipulate the order in which the transactions are executed to earn profit.
- *Timestamp Dependence*. The miners set the timestamp for the block they mined (generally according to the miner's local clock system). The miner can modify the timestamp by a few seconds on the premise that other miners accept the block they proposed. The vulnerability lies in the fact that some smart contracts take timestamp as a trigger condition, e.g., transfer money, adversary can thus manipulate the timestamp-dependent contracts for their own interests.
- *Mishandled exceptions*. Sometimes a contract (caller) calls another contract (callee), if callee runs abnormally, it terminates and return false. However, this exception may or may not be passed to the caller. In principle, the caller must explicitly check the return state from the callee to verify that the call was successfully executed. If the caller does not properly check the exception value, it will bring potential threats. A typical case is the KingOfTheEtherThrone (KoET) contract in Ethereum.
- *Reentrancy vulnerability*. When a contract calls another one, the current execution waits for the call to finish. As the fallback mechanism allow an attacker to re-enter the caller function, so attacker may use the intermediate state of the caller to conduct repeated calls, leading to loops of invocations which eventually consume all the gas. The most notorious reentrancy is The DAO attack.

4) Vulnerabilities. N. Atzei et al. conducted a systematic investigation on vulnerabilities of Ethereum smart contracts, which contains three level, namely Solidity, EVM bytecode, and blockchain [41]. Some representative vulnerabilities are detailed as follows:

- *Gasless send*. When a contract transfers some Ether to another contract through the **send** function, it may incur exception caused by out-of-gas. This vulnerability is subtle because Ether transfer is not generally associated with executing code.
- *Ether lost in transfer*. Currently, there are many orphan addresses on Ethereum, none of these addresses are associated with any user or contract. If a transaction is sent to an empty address, the Ether will disappear forever. Therefore, the user should pay attention to the validity of the recipient address. Besides, T. Chen et al. pointed out that the under-optimized smart contracts may also devour users' money [42].
- *Stack size limit*. Each time one contract invokes another, the call stack of the transaction increases by one frame. However, the total amount of call stack is limited (upper limit of the frame is 1024 for Ethereum). Once this limit is exceeded, an exception will be thrown [43].
- *Generating randomness*. Some specific smart contracts such as lotteries need generate pseudo-random number as a trigger condition. In general, the random number or its initialization seed is set as the hash (or the timestamp) of a particular future block. As we have already discussed, an adversary can manipulate the timestamp for profit [44].

5) Privacy leakage. Another issue that hinders the widespread use of smart contracts is the lack of privacy. All the action sequences of the smart contracts are recorded on the blockchain that are public to everyone. Even if users can use pseudonymous public keys to enhance privacy, the transaction values and account balances of public keys are still publicly accessible. For example, S. Meiklejohn et al. clustered Bitcoin addresses belonging to the same user, and used only a small

[15] Darkleaks. https://github.com/darkwallet/darkleaks

amount of transactions labeled through their empirical interactions with various services to identify major institutions [45]. D. Ron et al. analyzed the transaction graph of Bitcoin and obtain many statistical properties of user behaviors [46]. These privacy issues may cause deanonymization attacks against smart contracts.

6) Legal issues. Some scholars argue that smart contracts are merely a type of computer code that can self-execute, self-verify and self-constrain the performance of its instructions, which may represent all, part, or none of a valid legal contracts under the existing laws [47]. Thus, there may be a conflict between relational contract theories and smart contracts. For example, data privacy laws in European stipulate that citizens have a "right to be forgotten" which is incompatible with the immutable nature of blockchain-enabled smart contracts. Other legal issues include, but are not limited to:

- What laws otherwise apply to the transactions taking place within the smart contract application?
- What hazards are posed by use of the smart contract application alone (e.g., (i) a loss of data;(ii) business interruption; (iii) privacy breach; and/or (iv) a failure to perform)?
- What happens when the outcomes of the smart contract diverge from the outcomes that the law demands? [48-50]

B. Recent Advances

To deal with the above challenges, researchers have proposed several solutions. C. Natoli et al. [51] proposed to use functions such as **SendIfReceived** to enforce the execution order of transactions to avoid TOD vulnerability in Ethereum. A. Kosba et al. proposed a blockchain model called Hawk that does not store financial transactions clearly on the blockchain [52], thus retaining users' transaction privacy. For misalignment between the semantics of smart contracts platforms and the intuition of programmers, L. Luu et al. developed an open source security analysis tool called Oyente that leverages symbolic execution to discover potential security bugs in Ethereum smart contracts. They also proposed ways to enhance the operational semantics of Ethereum to make smart contracts less vulnerable. Among 19,366 smart contracts in Ethereum, Oyente flagged 8,833 of them vulnerable, including the The DAO [40]. To address the lacking of trustworthy data feeds, F. Zhang et al. introduced a system called Town Crier (TC) that acts as a reliable connection between HTTPS-enabled websites and smart contracts [37].

Another important way to improve the security of smart contracts is called formal verification, which provides a formal proof on an abstract mathematical model of the system. Formal verification of smart contracts involves proving that a contract program satisfies a formal specification of its behavior [53]. For example, Securify[16] is a web-based security analysis tool that uses formal verification and static analysis checks to identify vulnerabilities such as recursive calls and unexpected Ether flows. F* Framework [6] is a formal verification framework for analyzing the functional correctness and the runtime safety of Ethereum smart contracts by translating Solidity programs and EVM bytecode into F* (a functional programming language), and then identifying potential vulnerabilities. Remix [17] is a web-based IDE that relies on deductive program verification and theorem provers to search vulnerable coding patterns [54].

V. FUTURE TRENDS

The rapid development of the Internet and its deep coupling with the physical world have fundamentally changed the management pattern of modern organizations and societies. The future development trend of organizations/societies is bound to a transformation from Cyber-physical systems (CPS) to Cyber-physical-social systems (CPSS) in which social and individual factors must be taken into account [55]. At present, the concept of parallel societies based on CPSS has sprouted, and their substantive characteristics are Uncertainty, Diversity and Complexity (UDC) due to the engineering complexity and social complexity [56].

Blockchain and smart contracts are the infrastructure for implementing the CPSS-based parallel societies because they provide a set of effective decentralized data structures and interaction mechanisms for distributed social systems and distributed artificial intelligence [57]. As mentioned earlier, nodes running smart contracts can be seen as software agents who are the autonomous entities that have an understanding of external environment and act upon it. Since different nodes represent the interests of different individuals in an organization/society, they deploy and execute contracts through autonomous negotiation, thus forming various DAO/DAC/DAS. Beyond the traditional organizations/societies that organized in a hierarchical structures and top-down commands, DAO/DAC/DAS can help solve two main problems in organizational management, namely, Principal-Agent dilemma and high transaction costs of coordination.

The ACP approach (Artificial societies + Computational experiments + Parallel execution, where artificial systems are used for modeling and representation, computational experiments are utilized for analysis and evaluation, and parallel executions are conducted for control and management of complex systems) is by far the only systematic research framework in the field of parallel organizational/societies management [58]. Y. Yuan & F. Y. Wang proposed the conceptual framework, fundamental theory and research methodology of parallel blockchain [59]. We believe that the ACP approach can be naturally combined with blockchain to realize smart contracts-driven parallel organizations/societies management. First, the P2P network, distributed consensus, and incentive mechanism of the blockchain is the natural modeling

[16] Securify. https://securify.ch/
[17] Remix. https://remix.ethereum.org/

of a distributed system, each node will act as an autonomous agent and eventually constitute software-defined social systems (corresponding to artificial societies); Second, the programmable feature of smart contracts enable a variety of "WHAT-IF"-typed virtual experimental design, experimental scenarios deduction, and experimental results evaluation (corresponding to computational experiments), so that the agents can make optimal decision in a specific scenario; Finally, the combination of blockchain and IoT can generate a wide variety of smart assets, making it possible to connect physical world and virtual cyberspace. Through the virtual-real interactions and parallel evolution between the physical and artificial organizations/societies, the optimal organizational/societies management scheme can be obtained (corresponding to parallel execution) [60].

To summarize, the ACP-based smart contracts have the potential of realizing the transformation of CPSS into the intelligent and autonomous organizations/societies with such characteristics as Agility, Focus and Convergence (AFC) through software-defined systems and knowledge automation [56].

VI. Conclusion

With the increasing popularization and deepened applications of blockchain technology, the emerging smart contracts have become a hot research topic in both academia and industry. The decentralization, enforceability, and verifiability characteristics of smart contracts enable an agreement being executed between untrusted parties without the involvement of a trusted authority or central server. Thus, smart contracts are expected to revolutionize many traditional fields, such as financial, management, healthcare, etc. In this paper, we present a comprehensive overview of smart contracts, including their operational mechanism, main-stream platforms, and application fields. Specially, we propose a basic research framework of smart contracts based on a novel six-layer architecture. Then we discuss the open challenges standing ahead of smart contracts and the recent research progresses. Finally, the future development trends are discussed. The focus of this paper is to make a systematic overview of smart contracts and identify research gaps that need to be addressed in future studies.

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/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html

[2] N. Szabo, "The Idea of Smart Contracts," 1997. [Online]. Available: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html

[3] J. Stark, "Making sense of blockchain smart contracts," 2016. [Online]. Available: https://www.coindesk.com/making-sense-smart-contracts/

[4] 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.

[5] " Ethereum Yellow Paper," 2018. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[6] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, et al., "Formal verification of smart contracts: Short Paper," in *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security (PLAS '16)*, Vienna, Austria, 24 - 24 Oct., 2016, pp. 91-96.

[7] "What is a DAO," [Online]. Available: https://blockchainhub.net/dao-decentralized-autonomous-organization/

[8] G. W. Peters, and E. Panayi, "Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the Internet of money," arXiv:1511.05740v1, 2015.

[9] R. Qin, Y. Yuan, S. Wang, and F. Y. Wang, "Economic issues in bitcoin mining and blockchain research," in *2018 IEEE Intelligent Vehicles Symposium*, Changshu, Jiangsu, China, 26-29 June, 2018, in press.

[10] R. R. Bliss, and R. S. Steigerwald, "Derivatives clearing and settlement: A comparison of central counterparties and alternative structures," *Economic Perspectives*, vol. 30, no. 4, pp. 22-29, 2006.

[11] " How Ethereum smart contracts could revolutionize the financial securities industry," 2018. [Online]. Available: https://blog.polymath.network/how-ethereum-smart-contracts-could-revolutionize-the-financial-securities-industry-a683dacf1dae

[12] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and Victor Santamaria, "Blockchain and smart contracts for insurance: Is the technology mature enough?" *Future Internet*, vol. 10, no. 2: 20, 2018.

[13] M. Alharby, and A. V. Moorsel, "Blockchain-based smart contracts: A systematic mapping study," in *International Conference on Artificial Intelligence and Soft Computing*, 2017, pp. 125-140.

[14] W. Banasik, S. Dziembowski, and D. Malinowski, "Efficient zero-knowledge contingent payments in cryptocurrencies without scripts," in *European Symposium on Research in Computer Security*, Springer, Cham, pp. 261-280, 2016.

[15] A. Lielacher, "How blockchains can disrupt the mortgage market," 2017. [Online]. Available: https://www.nasdaq.com/article/how-blockchains-can-disrupt-the-mortgage-market-cm848889

[16] S. Wang, X. C. Ni, Y. Yuan, X. Wang, L. W. Ouyang, and F. Y. Wang, "A preliminary research of prediction markets based on blockchain powered smart contracts," in *2018 IEEE International Conference on Blockchain (Blockchain-2018)*, Halifax, Canada, 30 Jul. - 3 Aug., 2018, in press.

[17] J. L. de la Rosa, D. Gibovic, V. Torres-Padrosa, et al., "On intellectual property in online open innovation for SME by means of blockchain and smart contracts," in *Proceedings of the 3rd Annual World Open Innovation Conference (WOIC)*, Barcelona, Spain, 15 - 16 Dec., 2016.

[18] R. Beck, M. Avital, M. Rossi, and J. B. Thatcher, "Blockchain technology in business and information systems research," *Business & Information Systems Engineering*, vol. 59, no. 6, pp. 381-384, 2017.

[19] *Handbook on Business Process Management 1*, 2nd ed., Springer-Verlag Berlin Heidelberg, 2015, pp. 105-122.

[20] X. C. Ni, S. Zeng, X. Han, Y. Yuan, F. Y. Wang, "Organization management using software-defined robots based on smart contracts," in *2018 IEEE Intelligent Vehicles Symposium*, Changshu, Jiangsu, China, 26-29 June, 2018, in press.

[21] I. Weber, V. Gramoli, A. Ponomarev, et al., "On availability for blockchain-based systems," in *Proceedings of the IEEE International Symposium on Reliable Distributed Systems (SRDS'17)*, IEEE, 2017, pp. 64-73.

[22] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for Boardroom voting with maximum voter privacy," in *International Conference on Financial Cryptography and Data Security*, Springer, Cham, 2017, pp. 357-375.

[23] Q. Xia, E. B. Sifah, K. O. Asamoah, et al., "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain," in *IEEE Access*, vol. 5, pp. 14757-14767, 2017. doi: 10.1109/ACCESS.2017.2730843

[24] A. Azaria, A. Ekblaw, T. Vieira, et al., "MedRec: Using blockchain for medical data access and permission management," in *International Conference on Open and Big Data (OBD)*, Vienna, Austria, 2016, pp. 25-30. doi: 10.1109/OBD.2016.11

[25] T. T. Kuo, and L. Ohno-Machado, "ModelChain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks", arXiv:1802.01746 , 2018.

[26] K. Christidis, and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, 2016, 4: 2292-2303.

[27] A. Dorri, S. S. Kanhere, R. Jurdak, et al, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* , IEEE, 2017, pp. 618-623.

[28] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for IoT," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, ACM, 2017, 173-178.

[29] Y. Zhang, and J. T. Wen, "An IoT electric business model based on the protocol of bitcoin," in *2015 18th International Conference on Intelligence in Next Generation Networks (ICIN)*, Paris, France, 17-19 Feb., 2015, pp. 184-191.

[30] F. Knirsch, A. Unterweger, and D. Engel, "Privacy-preserving blockchain-based electric vehicle charging with dynamic tariff decisions," *Computer Science-Research and Development*, vol. 33, no. 1-2, pp. 71-79, 2018.

[31] M. Staples, S. Chen, S. Falamaki, et al., "Risks and opportunities for systems using blockchain and smart contracts," *Data61(CSIRO)*, 2017.

[32] H. M. Kim, and M. Laskowski, "Toward an ontology-driven blockchain design for supply-chain provenance," *Intelligent Systems in Accounting, Finance and Management*, vol. 25, no. 1, pp. 18-27, 2018.

[33] Rob O'Byrne, "How blockchain can transform the supply chain," 2017. [Online]. Available: https://www.logisticsbureau.com/how-blockchain-can-transform-the-supply-chain/.

[34] J. J. Li, Y. Yuan, S. Wang, F. Y. Wang, "Transaction queueing game in bitcoin blockchain," in *2018 IEEE Intelligent Vehicles Symposium*, Changshu, Jiangsu, China, 26-29 June, 2018, in press.

[35] 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.

[36] S. Zeng, X. C. Ni, Y. Yuan, F. Y. Wang, "A bibliometric analysis of blockchain research," in *2018 IEEE Intelligent Vehicles Symposium*, Changshu, Jiangsu, China, 26-29 June, 2018, in press.

[37] 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.

[38] G. Greenspan, "Why many smart contract use cases are simply impossible," 2016. [Online]. Available: https://www.coindesk.com/three-smart-contract-misconceptions/

[39] A. Juels, A. Kosba, and E. Shi, "The Ring of Gyges: Investigating the future of criminal smart contracts," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*, Vienna, Austria, 24 - 28 Oct., 2016, pp. 283-295.

[40] 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.

[41] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts," Cryptology ePrint Archive: Report 2016/1007. [Online]. Available: https://eprint.iacr.org/2016/1007, 2016.

[42] T. Chen, X. Li, X. Luo and X. Zhang, "Under-optimized smart contracts devour your money," in *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Klagenfurt, Austria, 2017, pp. 442-446. doi: 10.1109/SANER.2017.7884650

[43] K. Delmolino, M. Arnett, A. Kosba, et al., "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, pp. 79-94.

[44] X. Q. Li, P. Jiang, T. Chen, X. P. Luo, Q. Y. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, https://doi.org/10.1016/j.future.2017.08.020., 2017.

[45] S. Meiklejohn, M. Pomarole, G. Jordan, et al., "A fistful of bitcoins: characterizing payments among men with no names," in *Proceedings of the 2013 conference on Internet measurement conference (IMC '13)*, Barcelona, Spain, 23 - 25 Oct., 2013, pp. 127-140.

[46] D. Ron, and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *International Conference on Financial Cryptography and Data Security*, Springer, Berlin, Heidelberg, 2013, pp. 6-24.

[47] T. Swanson, "Great chain of numbers: A guide to smart contracts, smart property and trustless asset management," 2014. [Online]. Available: http://www.ofnumbers.com/the-guide/

[48] J. Caytas, "Blockchain in the U.S. Regulatory Setting: Evidentiary Use in Vermont, Delaware, and Elsewhere," *Columbia Science & Technology Law Review*, May 30, 2017. [Online]. Available: https://ssrn.com/abstract=2988363

[49] J. D. Hansen, and C. L. Reyes, "Legal aspects of smart contract applications," Perkins Coie's Blockchain Industry Group, White Paper, May 2017.

[50] B. Marino, and A. Juels, "Setting standards for altering and undoing smart contracts," in *International Symposium on Rules and Rule Markup Languages for the Semantic Web*, Springer, Cham, 2016, pp. 151-166.

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

[52] 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, 22-26 May, 2016, pp. 839-858.

[53] S. Amani, M. Bégel, M. Bortin, and M. Staples, "Towards verifying ethereum smart contract bytecode in Isabelle/HOL," in *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2018)*, Los Angeles, CA, USA, 8 - 9 Jan., 2018, pp. 66-77.

[54] A. Dika, "Ethereum smart contracts: Security vulnerabilities and security tools," M.S. thesis, Applied Computer Science., Norwegian University of Science and Technology, Trondheim, Norway, 2017.

[55] 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.

[56] 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.

[57] Y. Yuan, T. Zhou, A. Y. Zhou, Y. C. Duan, and F. Y. Wang, "Blockchain technology: From data intelligence to knowledge automation," *Acta Automatica Sinica*, vol. 43, no. 9, pp. 1485-1490, 2017.

[58] 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.

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

[60] F. Y. Wang, Y. Yuan, C. Rong and J. J. Zhang, "Parallel blockchain: An architecture for CPSS-Based smart societies," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 303-310, 2018.