In December 2017, "CryptoKitties", a game on the Ethereum blockchain became an instant success shortly after its launch. It attracted 180,000 users with over $20 million of spend in Ether, and was at one point taking up 12% of all Ethereum transactions.

Technically speaking, "CryptoKitties" is a smart contract - a piece of code with storage capability that resides on a blockchain. Smart contracts are gaining increasing popularity in recent years. We present a comprehensive review of smart contracts with a focus on existing applications and challenges they face. We have covered the smart contract mechanisms, promising use cases, as well as relevant research work and the open issues.

Additional Key Words and Phrases: Smart contracts, Blockchain

1 INTRODUCTION

1.1 What Are Smart Contracts?

The concept of smart contracts was introduced in the 1990s by Wei Dai. In a post on anonymous credit, Wei described an anonymous loan scheme with redeemable bonds and lump-sum taxes to be collected at maturity. Unlike traditional contracts that rely on the reputation of the counterparties, smart contracts can be made between untrusted, anonymous people. Also, the execution of contractual terms is automatic and does not rely on any third party. Szabo et al. [23] later discussed the potential form of smart contracts and proposed to use cryptographic mechanisms to enhance the security.

Smart contracts have become a reality with the boom of blockchain technology, which operates without trusted third parties for settling transactions and disagreements among pseudonymous participants. The first successful implementation was Bitcoin Script, a purposely not-turing-complete language with a set of simple, pre-defined commands. Similarly, Nxt, a new generation of public blockchain, provides users with several application templates, including Decentralized Asset Exchange, Marketplace, and Voting. In addition, there are also platforms that enable more complex functionalities and flexibilities, e.g., Ethereum [24], which has adopted turing-complete languages for smart contracts. Nowadays, smart contracts are also being built on consortium blockchains, e.g., the Hyperledger Fabrics. Figure 1 shows the evolution of smart contracts.

1https://www.cryptokitties.co
2https://en.bitcoin.it/wiki/Script
3https://nxtplatform.org
4https://www.hyperledger.org
1.2 Why Smart Contracts?

Firstly, blockchains and other distributed ledgers can maintain an immutable record of data and effectively mitigate single points of failure. Secondly, since smart contracts inherit the encryption pseudonymity of blockchains, even with the code and data accessible to everybody, smart contracts are able to prevent unwanted monitoring and tracking. Thirdly, thanks to the flexibility of programming languages, smart contracts also have a good interoperability among multiple instances and are able to tackle modifications better [11].

Therefore, smart contracts have the potential to reduce risks, improve efficiency and save costs, when it comes to transaction settlement among different parties. Unlike traditional paper contracts that rely on middlemen and third-party intermediaries for execution, smart contracts help to automate contractual procedures, minimize interactions between parties, and reduce the cost of administration and service. Smart contracts on public blockchains (public smart contracts) appear to be attractive for start-ups and people in academia, while smart contracts on permissioned blockchains (permissioned smart contracts) have the potential to speed up transaction settlement, reduce risk and costs in business consortiums.

| Common                  | Public Smart Contracts                                                                 | Permissioned Smart Contracts                          |
|-------------------------|----------------------------------------------------------------------------------------|-------------------------------------------------------|
|                         | Immutable record                                                                       |                                                      |
|                         | Proper encryption on data and pseudonymity                                             |                                                      |
|                         | Interoperability among different platforms                                             |                                                      |
|                         | Traceable modifications                                                                 |                                                      |
| Unique                  | Easy to deploy                                                                         | Lower settlement                                      |
|                         | Accessible for the public                                                               | Lower operational cost                                |
|                         |                                                                                        | Permissioned access                                   |

Table 1. Characteristics of public and permissioned smart contracts.

Despite the hype of smart contracts, the technology is still in its infancy and there is a long way to go before its full adoption. This paper explores the difference between public and permissioned smart contracts, provides examples of existing smart contract applications and highlights the challenges to overcome for the popularization of the technology.
2 SMART CONTRACT MECHANISMS

2.1 Overview

The operation of smart contracts can hardly be decoupled from the underlying blockchain, which is originally known as the "technology behind Bitcoin [18]", the first successful cryptocurrency that enables peer-to-peer (P2P) transaction settlement without going through any financial institutions. State of a blockchain is represented by a series of transactions and is computed after every block [3]. Transactions can be automatically triggered when certain conditions are met, and this is achieved with smart contracts that allow users to form mutual agreements with each other. As blockchains have storage capability, the concept has been broadened from simple P2P transaction settlement, to what is known as the "distributed ledger technology" or "DLT".

As the expectation and requirements of smart contracts are often different for public and permissioned blockchains, we below discuss the two separately.

2.2 Public Smart Contracts

Public smart contracts set no restrictions for peers to deploy or use. Although Bitcoin Script allows users to construct simple smart contracts, the general-purpose Ethereum contracts have quickly overtaken Bitcoin Script in popularity. Based on the statistics in Etherscan, there are more than 2 million Ethereum accounts holding 96.5 million of Ethers. Half of them are contract accounts, with the total value being 12 million Ether. Competitors such as Neo and EOS are also independent blockchains facilitating peer consensus and smart contracts. To understand the popularity of different platforms, we obtained the number of smart contract projects deployed on Github from the beginning of 2015 till the end of 2017. As illustrated in Figure 3, Ethereum is the most popular platform among the 7 blockchain instances we surveyed.

To give readers an intuitive idea of how smart contracts work on public blockchains, we below use Ethereum as an example. Just like most other popular blockchain platforms, Ethereum uses the proof-of-work (PoW) mining protocol for network consensus. Ethereum smart contracts reside in Ethereum Virtual Machines (EVMs), which isolates them from the blockchain network, and prevents the code running inside an EVM from accessing the filesystem and other processes. Once deployed, the smart contract obtains a unique address that is linked to a balance, similar to an externally controlled account (EOA) that is often owned by a user. A smart contract can send transactions to an EOA or another contract.

Figure 2 illustrates the working of Ethereum smart contracts, where the mining process is omitted for simplification. In the first step, Client 1 creates a smart contract for voting in a high-level language, e.g. Solidity. This smart contract is compiled into machine-level byte code where each byte represents an operation, and then uploaded to the blockchain in the form of a transaction by EVM 1. A miner picks it up and confirms it in Block #i+1. Once a voter has submitted his vote via the web interface, the EVM 2 queries the data from the web and embeds it into Transaction tx and deploy it to the blockchain. State of the voting contract is updated in Block #i+2 with the confirmation of transaction tx. If Client 3, the coordinator, later wants to check the states stored in the contract, s/he has to synchronize up to at least Block #i+2 to see the changes caused by tx.

---

5. https://etherscan.io/accounts/c  
6. https://neo.org  
7. https://eos.io  
8. https://github.com  
9. https://en.bitcoin.it/wiki/Comparison_of_cryptocurrencies  
10. https://solidity.readthedocs.io/en/develop/
On public blockchains, fees are often charged to deploy and update the states stored in smart contracts as the computation is performed by all network nodes and these fees can be high. For example, based on a study by Rimba et al. [20], the cost of maintaining smart contracts on the public Ethereum blockchain is nearly 360 times higher than using business models deployed on Amazon SWF.

2.3 Permissioned Smart Contracts

In recent years, the concept of blockchain has been significantly broadened by banks and many other industries. “Permissioned blockchains”, including both “private blockchains” and “consortium blockchains” are widely used in businesses. There is a significant difference between public and permissioned blockchains. Unlike the global-level, fully-untrusted, public blockchains on public blockchains, fees are often charged to deploy and update the states stored in smart contracts as the computation is performed by all network nodes and these fees can be high. For example, based on a study by Rimba et al. [20], the cost of maintaining smart contracts on the public Ethereum blockchain is nearly 360 times higher than using business models deployed on Amazon SWF.

2.3 Permissioned Smart Contracts

In recent years, the concept of blockchain has been significantly broadened by banks and many other industries. “Permissioned blockchains”, including both “private blockchains” and “consortium blockchains” are widely used in businesses. There is a significant difference between public and permissioned blockchains. Unlike the global-level, fully-untrusted, public blockchains on public blockchains, fees are often charged to deploy and update the states stored in smart contracts as the computation is performed by all network nodes and these fees can be high. For example, based on a study by Rimba et al. [20], the cost of maintaining smart contracts on the public Ethereum blockchain is nearly 360 times higher than using business models deployed on Amazon SWF.

2.3 Permissioned Smart Contracts

In recent years, the concept of blockchain has been significantly broadened by banks and many other industries. “Permissioned blockchains”, including both “private blockchains” and “consortium blockchains” are widely used in businesses. There is a significant difference between public and permissioned blockchains. Unlike the global-level, fully-untrusted, public blockchains on
which voluntary miners exhaust their computational resources on PoW puzzles, permissioned blockchains are semi-trusted and hence usually do not require replication for network consensus. Despite the alteration from the original blockchain concept, there exists a large number of industrial applications where a permissioned blockchain is used to manage the collaboration of multiple business partners. To avoid the inefficient consensus protocols, a number of technology companies are designing their own blockchain platforms to support semi-trusted consortiums.

As an example, the Linux Foundation is developing Hyperledger for business processes and collaborations that involve multiple parties. It reduces the cost of consensus by eliminating PoW mining and implementing a Practical Byzantine Fault-tolerant (PBFT) protocol [5] instead. Transaction ordering is handled by an “orderer” that collects transactions and takes votes from selected peers. The “orderer” groups validated transactions into a block and sends them as a batch to all other peers once ready. The block size can be defined in either number of transactions or time of waiting. Chaincode [12] is the equivalence of smart contracts in Hyperledger. Although free from mining, all network nodes still have to execute all transactions and smart contracts individually for synchronization. As an additional property, secure channels in Hyperledger allow a group of participants to create a separate ledger of transactions without acknowledging transactions to other groups.

To reduce the burden of blockchain nodes, some suggest that complex business logics should be moved to a separate middle layer instead of the blockchain itself. For instance, Microsoft Azure is developing Cryptlets, [13] where a central host execute smart contracts to support the separation of data and logic on consortium blockchains.

3 SMART CONTRACT APPLICATIONS

Public blockchains facilitate the convenient developing and testing of smart contract applications, which is favored by many startups who raise funds through Initial Coin Offerings (ICOs). Big enterprises on the other hand, mainly want to take the advantage of permissioned blockchains and customised smart contracts to replace their traditional bussiness models. Some of the popular use cases include: banking, Electronic Medical Record (EMR), IoT data management [6], in addition, there are also other interesting applications such as smart waste management, real estate, and ride-sharing arcade city as summarised in Table 2. We below present a survey of existing smart contract applications and discuss their strengths and weaknesses as well as the potential of having wider adoption.

3.1 Public Smart Contracts

Health Care and Medical Records. Besides games, gambling and ICOs, a major application is related to health care and access control of medical records. Blockchain technology and smart contracts are seen by many health care professionals as a secure way of sharing and accessing patients’ EMR. Smart contracts can feature multi-signature approvals between patients and providers to only allow authorised users or devices to access or append the record. They also enable interoperability via collaborative version control to maintain the consistency of the record. Besides bene\footnote{http://hyperledger-fabric.readthedocs.io/en/release/chaincode.html}\textit{\textsuperscript{12}}\textit{\textsuperscript{12}}\textsuperscript{12}\textsuperscript{12}f\textit{\textsuperscript{14}}\textit{\textsuperscript{14}}\textsuperscript{14}\textsuperscript{14}iting patients and their care providers, smart contracts can also be used to grant researchers access to certain personal health data and enable micropayments to be automatically transferred to the patients for participation [14].

\footnote{http://hyperledger-fabric.readthedocs.io/en/release/chaincode.html}
\footnote{https://github.com/Azure/azure-blockchain-projects/blob/master/bletchley/bletchley-whitepaper.md}
\footnote{https://www.healthit.gov/sites/default/files/11-74-ablockchainforhealthcare.pdf}
However, the realization of these applications is limited by the immature infrastructure of most public blockchains and high development costs. There are also concerns about policies and users’ willingness to publicise their personal information.

**Identity Management.** uPort \(^{15}\) is an identity management framework that leverages Ethereum smart contracts to recover accounts and protect user privacy in the case of a device loss. The main component - uPort identifier is a unique 20-byte hexadecimal string representing the address of a Proxy contract that lies in-between a controller contract and an application contract. uPort enables users to replace their private key (saved off-chain) while maintaining an on-chain persistent identifier. If a valid user brings a new device, s/he can seek for approval from a list of existing recovery delegates, and replace the old user address with a new one.

Identity management frameworks using blockchain still need to go through a number of enhancements before adoption. In the case of uPort, the publicity of the recovery delegates of a user poses the security risk of compromising user identities.

**Scaling Blockchains.** Although the fundamental limits in the expressiveness of Bitcoin Script make the retrofitting time-consuming and costly \(^{8}\), it helps to prevent malicious contracts and safeguard the system. Bitcoin is now developing the Lightning Network \(^{19}\) through smart contracts to facilitate off-chain transactions using payment channels. The goal is to improve the scalability of this global system by reducing the on-chain verification and storage. The corresponding implementation in Ethereum is the Raiden Network.\(^{16}\)

### 3.2 Permissioned Smart Contracts

**Banking.** Banking remains to be a main area for various smart contract use cases. One example is the mortgage service. According to a report made by Capgemini Consulting \(^{4}\), with smart contracts in mortgage, consumers could potentially save US$480-960 per loan and banks would be able to cut US$3 billion-$11 billion of annual costs in the US and Europe. Banks can also use smart contracts to streamline clearing and settlement processes. It has been reported that more than 40 global banks have participated in a consortium to test smart contracts for clearing and settlement activities.\(^{17}\) In addition, the know your customer (KYC) and anti money laundering (AML) policies can also be embedded easily with the smart contract logic.

However, the interoperability with legacy systems and the scalability of blockchains remain to be obstacles in realising such systems. Also, it is crucial that the smart contract implementation is secure against attacks that are aimed at stealing of assets or tampering of the contract code \(^{2}\).

**Provenance & Supply Chain.** Blockchain can help solve some of the key problems in the supply chain and logistics such as transparency, optimization, security and visibility of various operations in the transportation of goods \(^{21}\). A supply chain with continuous, real-time access to reliable, shared data is more efficient than traditional supply chains. Provenance of the product via the blockchain also raises the bar on quality in production by reducing the risk of wastage and spoilage. In addition, the blockchain itself helps to ensure trust among different parties. An example use case is the IBM’s blockchain-based food supply chain. Compared to public blockchains, a consortium blockchain would make more sense in this context as it reduces the cost and accelerates transaction processing.

Despite the advantages of using blockchains in supply chains, the integration of blockchains with existing platforms and business processes is still in the early stages. The use of smart contracts

---

\(^{15}\)https://www.uport.me

\(^{16}\)https://raiden.network

\(^{17}\)http://blockchainapac.fintecnet.com/uploads/2/4/3/8/24384857/smart_contracts.pdf
to negotiate and finalize transactions may require major changes in the supply chain workflow. Apart from that, resistance from the banks, exchange networks or trusted intermediaries could also delay the blockchain adoption.

**Voting.** Voting is another application that can benefit from blockchain and smart contracts. A Danish political party has implemented a smart contract to ensure the fairness and transparency for internal election purposes.\(^{18}\) Mccorry et al. [15] proposed a boardroom voting scheme that is different from existing proposals of e-voting. Mccorry’s system works under the assumption of a small group of voters with known identities and provides maximum voter privacy and verifiability. Mccorry et al. have also tested the system’s feasibility on a Ethereum private network and estimated the cost of $0.73 per voter for running it. The statistics have shown that public blockchains are more feasible for small polls whereas permissioned blockchains will be required to run national scale elections.

**IoT.** A promising but controversial application scenario is the use of blockchain and smart contracts for IoT data management. Intuitively, as both systems are decentralised in nature, blockchains could be used to enhance trust in IoTs that are constantly sharing and exchanging a large amount of data. However, the other properties of blockchain and IoT do not seem to fit naturally together. Firstly, IoT data is often sensitive, and should not be shared with everyone else. Secondly, blockchains are resource-consuming. Even with lighter consensus mechanisms, the idea of having all IoT devices to execute all programs seems redundant considering their limited processing capability.

As a major player in the field, IBM is integrating the Watson IoT Platform with the IBM Blockchain built on top of Hyperledger Composer. The goal is to build a trusted, low-cost and efficient business network while maintaining an indelible record to satisfy industrial and governmental requirements. Similarly, Chain of Things\(^ {19}\) is also trying to merge blockchain with IoT to achieve security, reliability and interoperability.

**Insurance.** In the insurance industry, smart contracts can perform error checking, routing, approve workflows, and calculate payouts based on the type of claim and the underlying policy. For example, the processing of travel insurance claims can be automatically verified against flight delays or cancellations. Smart contracts can help to remove the human factor involved in the process, therefore decreasing the overall administrative cost for the insurers and increasing the transparency for the consumers\(^ {4}\).

Nonetheless, technological limitations and legal regulations are major challenges to be addressed before shifting to smart contracts for insurance policies. Another drawback is the inflexibility of smart contracts. Traditional contracts can be amended or terminated upon agreement between both parties, but smart contracts as computer programs have no such mechanism. Moreover, more authorities are needed to recognize the legality of financial smart contacts.

### 4 RESEARCH AND OPEN CHALLENGES

Although the potential of smart contracts is tremendous, most platforms and applications are still in their infancy and come with a handful of problems, ranging from a lack of regulation to the pseudonymous operation of criminal activities. In this section, we summarize the limitations of

\(^{18}\)http://www.newsbtc.com/2014/04/22/danish-political-party-may-first-use-block-chain-internal-voting

\(^{19}\)https://www.chainofthings.com

\(^{20}\)http://variableauto.com/wp-content/uploads/2017/01/Private-BlockChain-in-automotive-Safety.pdf

\(^{21}\)http://scet.berkeley.edu/wp-content/uploads/AIR-2016-Blockchain.pdf
smart contracts and solutions proposed in recent research works, we also discuss the remaining challenges and provide insights on future directions. Four main topics are covered, namely, Technology, Legalization, Acceptance, and Usability. A summary of applications and challenges associated with them can be found in Table 3.

4.1 Technology
We discuss below the weak links and challenges in the composition and execution of smart contracts from a technical perspective. Note that we here only provide a limited number of examples, a more detailed mapping study on various issues of smart contracts can be found in [1].
4.1.1 Security. Security is one of the major concerns of any blockchain system and related procedures. In 2016, a re-entrancy attack in Solidity caused a loss over $40M and has led to a heated discussion over security issues of Ethereum smart contracts. In fact, many vulnerabilities are caused by the misunderstanding of the scripting languages [2].

**Current Research:** Following the work by Juels et al. [12] in which several forms of criminal Ethereum smart contracts were explored, Luu et al. [14] further studied security flaws of existing Ethereum smart contracts including how contract execution and code behaviour are affected by the order of mined transactions, correctness of timestamps and handling of exceptions. Luu et al also developed OYENTE, to analyse and detect security-related document bugs of smart contracts and proposed a set of improvements to the Ethereum protocol to enhance security. Delmolino et al. summarized common mistakes students made while programming smart contracts in the Serpent language [8]. Apart from not realizing the limitation of the blockchain implementation, students often fail to encode state machines logically and ensure the incentive compatibility of a contract. Some other groups are also developing alternatives. As an example, Coblenz et al. are working on the Obsidian coin [7] which comes with a new programming language to enhance the security and usability of smart contracts.

**Remaining Gaps:** As majority of the technical flaws are related to the semantic of scripting languages, there is an urgency to improve existing scripting languages and develop alternatives. An efficient way to conduct excessive testings and find flaws is to make things open-source and obtain feedback from the community as what is being done by [18, 24]. Also, since the types of attacks vary from platform to platform, there is a need to understand the mechanism and vulnerabilities of a particular blockchain before using it.

4.1.2 Privacy. Public Smart contracts are transparent to all participants and therefore susceptible to privacy leakage, but the intention of keeping secrets is to some extent contradictory to the nature of public blockchains and hence requires additional operations.

**Current Research:** One way to protect privacy is to integrate an extra component for data protection, i.e. the Zero-Knowledge Proofs (ZKP) in ZeroCoin [17]. Similar ideas and techniques can also be applied to smart contracts. In Hawk [13], a privacy-preserving compiler was built on top of the ZeroCoin protocol to enable the compilation of smart contracts with a cryptographic protocol while maintaining users’ on-chain privacy and contractual security. With a minimally-trusted manager who execute the code, two users can perform actions on smart contracts without revealing the actual information.

**Remaining Gaps:** The adoption of encryption algorithms may cause extra burden for the system.

4.1.3 Integrity. Even though the execution of smart contracts is regulated by hard-coded software programs and handled by all network participants, the data fed to smart contracts is still controled by outside parties and cannot be fully trusted.

**Current Research:** Town Crier by Zhang et al. [25] serves as a bridge between smart contracts and popular websites to secure the data-delivery. Deployed on the Intel Software Guard Extensions (SGX) hardware that provides a secure enclave for software processing, Town Crier can reliably fetch data from trusted websites to blockchain smart contracts.

**Remaining Gaps:** Town Crier ensures the integrity of data fed towards smart contracts, but not the users. In most cases, users cannot directly access data on a blockchain or smart contract. Instead, they do so via wallet apps developed by other parties, which makes data integrity out of users’ control.
4.2 Legalization
Before smart contracts are ready to experience a wider adoption and reshape business procedures, many fundamental issues are to be solved. Notably, there is still lack of a formalised way of composing smart contracts to suit various design purposes, especially when legal contents are involved. From a legal perspective, there is lack of regulation and policies on smart contracts. It is sometimes hard for blockchains and smart contracts to obtain government approval. By now there is still the issue of enforceability and jurisdiction with this technology. When evaluating opportunities, organizations should carefully consider the effect of such lack of government acceptance.

**Current Research:** In the case of Solidity, Frantz et al. [9] have proposed a reasonable way of mapping contractual semantics to software declarations that covers the 5 essential components, i.e. “Attributes”, “Deontic”, “Aim”, “Conditions” and “Or else” (or “ADICO”). According to the authors, to successfully convert between institutional constructs and smart contracts, both directions need to be taken into consideration [9].

**Remaning Gaps:** Scripting languages need to be regulated in a way to be more comprehensive and easy-to-use for both technical and non-technical people.

4.3 Acceptance
Despite the hype of blockchains and smart contracts in both public and consortium domains, there are still a number of misconceptions about the technology. Firstly, there have been an inflated expectation and many unrealistic use cases. Secondly, even with proper use cases, it can be hard to persuade stakeholders and users to accept the new technology. This could result in extra development costs and a low return on the investment.

**Remaning Gaps:** Some of the proposed use cases are in fact more efficient to be implemented via traditional databases. Hence, those who are interested in developing smart contract applications should keep in mind what can be achieved and what can not with it, together with the development cost.

4.4 Usability
Although being called “smart contracts”, these logic-based computer programs have a limited level of interactivity and do not allow people to negotiate and make changes based on the later agreed modifications like in traditional contracts, and they are not flexible with exceptions such as glitches. Also, due to the P2P nature of blockchains, letting ordinary users control their data directly is risky, and the exchange rate can be unpredictable when cryptocurrencies are involved.

**Remaning Gaps:** If the above-mentioned aspects can be properly handled without sacrificing the reliability and enforceability, smart contracts can expect to gain more popularity.

5 CONCLUSION
Smart contracts have the potential to solve existing problems in business procedures and are being developed and tested by many. However, smart contracts are still limited in their ability to fulfill all expectations, they need to be improved further in many different aspects. We believe the future development should mainly focus on improving the scripting languages, integration with existing procedures, as well as the usability and legality of smart contracts. If smart contracts can be made to work with enhanced security, legality and flexibility, we can foresee a wider adoption of smart contracts in both commercial and non-commercial domains.
REFERENCES

[1] Maher Alharby and Aad van Moorsel. 2017. Blockchain-based Smart Contracts: A Systematic Mapping Study. arXiv preprint arXiv:1710.06372 (2017).

[2] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. 2017. A Survey of Attacks on Ethereum Smart Contracts (SoK). In International Conference on Principles of Security and Trust, Springer, 164–186.

[3] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A Kroll, and Edward W Felten. 2015. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In Security and Privacy (SP), 2015 IEEE Symposium on. IEEE, 104–121.

[4] B Cant, A Khadikar, A Ruiter, JB Bronebakk, J Coumaros, J Buvat, and A Gupta. 2016. Smart Contracts in Financial Services: Getting from Hype to Reality. Capgemini Consulting (2016).

[5] Miguel Castro and Barbara Liskov. 2002. Practical Byzantine fault tolerance and proactive recovery. ACM Transactions on Computer Systems (TOCS) 20, 4 (2002), 398–461.

[6] Konstantinos Christidis and Michael Devetsikiotis. 2016. Blockchains and smart contracts for the internet of things. IEEE Access 4 (2016), 2292–2303.

[7] Michael Coblenz. 2017. Obsidian: A safer Blockchain programming language. In Proceedings of the 39th International Conference on Software Engineering Companion. IEEE Press, 97–99.

[8] Kevin Delmolino, Mitchell Arnett, Ahmed Kosba, Andrew Miller, and Elaine Shi. 2016. 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, 79–94.

[9] Christopher K Frantz and Mariusz Nowostawski. 2016. From institutions to code: towards automated generation of smart contracts. In Foundations and Applications of Self* Systems, IEEE International Workshops on. IEEE, 210–215.

[10] Ye Guo and Chen Liang. 2016. Blockchain application and outlook in the banking industry. Financial Innovation 2, 1 (2016), 24.

[11] Florian Idelberger, Guido Governatori, Régis Riveret, and Giovanni Sartor. 2016. Evaluation of logic-based smart contracts for blockchain systems. In International Symposium on Rules and Rule Markup Languages for the Semantic Web, Springer, 167–183.

[12] Ari Juels, Ahmed Kosba, and Elaine Shi. 2016. The ring of Gyges: Investigating the future of criminal smart contracts. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 283–295.

[13] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. 2016. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In Security and Privacy (SP), 2016 IEEE Symposium on. IEEE, 839–858.

[14] Lai Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. 2016. Making smart contracts smarter. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 254–269.

[15] Patrick McCorry, Siamak F Shahandashti, and Feng Hao. 2017. A Smart Contract for Boardroom Voting with Maximum Voter Privacy. IACR Cryptology ePrint Archive 2017 (2017), 110.

[16] Esther Mengelkamp, Johannes Gärttner, Kerstin Rock, Scott Kessler, Lawrence Orsini, and Christof Weinhardt. 2018. Designing microgrid energy markets: A case study: The Brooklyn Microgrid. Applied Energy 210 (2018), 870–880.

[17] Ian Miers, Christina Garman, Matthew Green, and Aviel D Rubin. 2013. Zerocoin: Anonymous distributed e-cash from bitcoin. In Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 397–411.

[18] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[19] Joseph Poon and Thaddeus Dryja. 2015. The bitcoin lightning network: Scalable off-chain instant payments. Technical Report (draft) (2015).

[20] Paul Romba, An Binh Tran, Ingo Weber, Mark Staples, Alexander Ponomarev, and Xiwei Xu. 2017. Comparing Blockchain and Cloud Services for Business Process Execution. In Software Architecture (ICSA), 2017 IEEE International Conference on. IEEE, 257–260.

[21] Krystsina Sadouskaya et al. 2017. Adoption of Block Chain Technology Supply Chain and Logistics. (2017).

[22] Jianjun Sun, Jiaqi Yan, and Kem ZK Zhang. 2016. Blockchain-based sharing services: What blockchain technology can contribute to smart cities. Financial Innovation 2, 1 (2016), 26.

[23] Nick Szabo. 1997. Formalizing and securing relationships on public networks. First Monday 2, 9 (1997).

[24] Gavin Wood. 2014. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper 151 (2014).

[25] Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. 2016. Town crier: An authenticated data feed for smart contracts. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 270–282.
| Description                                                                 | Security | Privacy | Integrity | Solidity | Compliance and intervention | Conceptual misalignment and positive feedback | Legitimacy | Liability | Development | Technology | Legacy | Interoperability with legacy systems | Usability |
|-----------------------------------------------------------------------------|---------|---------|-----------|----------|-----------------------------|-----------------------------------------------|-----------|-----------|-------------|------------|--------|------------------------------------|----------|
| Introduction of new direct ventures and security vulnerabilities.          |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Security                                                                     |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Lack of secrecy in contract execution and user information.                 |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Privacy                                                                     |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Weak access to contract information.                                        |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| There's no guarantee on the data integrity.                                |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Scenarios of fraud has been defined and accepted for each application.     |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Should be more actively involved in the technology.                         |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Ability of stakeholders to participate in the technology.                   |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Different opinions on the use of smart contracts across stakeholders.       |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| The technology and legislation frameworks should be developed in parallel.  |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Nearly all smart contracts are completed before the technology is developed.|         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Value proposition, extra development costs                                 |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Extra budget on deployment with low return of investment.                   |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Most ordinary users are not aware of the new technology and not willing to   |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| participate.                                                                |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| Smart contracts are complex programs that require deep knowledge and         |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| potential financial risks.                                                   |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |
| The smart contracts should work with current systems for smooth transition. |         |         |           |          |                              |                                               |           |           |             |            |        |                                    |          |

Table 3. Challenges [6, 10, 16, 22]