Smart Contract Development Model and the Future of Blockchain Technology

Richard Richard
Information Systems Department, School of Information Systems, Bina Nusantara University, Jakarta, Indonesia; Computer Science Department, BINUS Graduate Program - Doctor of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Harjanto Prabowo
Computer Science Department, BINUS Graduate Program - Doctor of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Agung Trisetyarso
Computer Science Department, BINUS Graduate Program - Doctor of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Benfano Soewito
Computer Science Department, BINUS Graduate Program - Doctor of Computer Science, Bina Nusantara University, Jakarta, Indonesia

ABSTRACT

Blockchain has become a mainstream technology in our society in recent years. With its nature of secure decentralization, people can create decentralized applications by developing smart contracts on top of a blockchain platform. With blockchain, technology is still in the developing phase, the smart contract development process in blockchain has its unique complexity and uncertainty. The condition will drive the challenge for any developers to work on this issue. In this research, we determine the smart contract development model in creating decentralized applications. Our proposed model aligned with our findings in the systematic mapping process of this study.

CCS CONCEPTS

• **Computer systems organization**; • **Architectures**; • **Distributed architectures**; • **Peer-to-peer architectures**; • **Computing methodologies**; • **Distributed computing methodologies**; • **Distributed programming languages**;

KEYWORDS

Blockchain, Decentralized Applications, Smart Contract, Smart Contract Development, Systematic Mapping

ACM Reference Format:
Richard Richard, Harjanto Prabowo, Agung Trisetyarso, and Benfano Soewito. 2020. Smart Contract Development Model and the Future of Blockchain Technology. In *2020 the 3rd International Conference on Blockchain Technology and Applications (ICBTA 2020), December 14–16, 2020, Xi'an, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3446983.3446994

1 INTRODUCTION

A smart contract in blockchain technology is one of the critical functions driving blockchain adoption in different sectors. The original concept of a smart contract is a computer protocol that digitally facilitates, verifies, and enforces the contract's performance [30]. With a smart contract, we may enforce autonomous verification in any transaction in the network under some pre-defined conditions. In using this autonomous verification, the needs of intermediaries in the network may be very reduced. All parties involved should trust each other using specific rules written in the Smart Contract [28]. In the case of real-world use of smart contracts, smart contracts may be used in any sector that uses the agreement as a base process. Subsequently, a smart contract in blockchain technology allows the parties in a blockchain network to benefit from an autonomous verification concept. The smart contract is stored in a decentralized blockchain system, and every transaction is verified by all the nodes on the blockchain network [6]. Nowadays, smart contract capabilities are becoming a key feature to be implemented in a blockchain network.

The marriage of blockchain technology and smart contracts is first known to the general public through the Ethereum project launched by Vitalik Buterin with his first white paper on Ethereum [3]. In Ethereum blockchain, a smart contract is compiled and executed by Ethereum Virtual Machine (EVM) [10], a distributed extensive virtual machine system. Besides, a blockchain network's smart contract capability allows users to build any application running on the blockchain network called Decentralized Applications (DApps). Often driven by the success of the ERC-20 contract standard, which offers a programming interface for token formation through a smart contract, is the further adoption of smart contracts in blockchain. This token development feature increases blockchain's popularity with a crowdfunding-like process called Initial Coin Offering (ICO) around 2016 through 2018 [8]. DApps and ICO phenomena leave a mixed impression related to the future of blockchain technology. Many issues emerged, including smart contract protection, interoperability, scalability, and contract continuity. Know Your Customer (KYC), and Anti Money Laundering (AML) compliance have also become an important issue linked to the very high number of ICO Scams in 2017 [11]. Not to mention the lack of user experience

ICBTA 2020, December 14–16, 2020, Xi'an, China

ACM ISBN 978-1-4503-8896-2/20/12…$15.00
https://doi.org/10.1145/3446983.3446994

with the current decentralized application due to the complexity of the blockchain system architecture. These issues urge the parties involved to develop a standard that could be used in the smart contract development process

The terminology of decentralized applications (DApps) is defined as a broader definition of smart contracts. Decentralized applications in blockchain technology do not have a truly central failure point and, therefore, have advantages for data security, privacy, and user data ownership [26]. The representation of DApps is fueled by using a smart contract, which has been configured with a set of rules and serves as governance in the Dapps. Smart contract work as an immutable and transparent program that is stored and executed within a blockchain platform. As a result, the software development process involving a smart contract (DApps) is quite different from the usual software development process [20]. It is even more difficult than ever in a real-world situation as the smart contract moves forward towards software scalability and continuity. Parizi et al. [24] carried out an empirical review of several smart contract programming languages focused on usability and security for a new smart contract developer. Unfortunately, the result shows that when it comes to security vulnerabilities in the smart contract implemented by new developers, Solidity, one of the most popular programming languages in the smart contract, is falling short behind as we found it most vulnerable to security vulnerabilities. While being usable is a huge plus, on the other hand, being vulnerable to security vulnerabilities is a huge downside, as these security vulnerabilities can be exploited by malicious users to cause financial damage, as seen from recent attacks on the platform. This security problem was exploited in April 2020 when hackers just tapped China's dForce for $25 million in Ethereum ERC777 vulnerability exploit [14]. The Block, online research, review, and newsroom for digital assets, found that the word "compound" was four times founded within the Smart Contract of dForce. This problem gives rise to the perception that the dForce team cannot create a smart contract of its own.

This paper focused on reviewing current smart contract development method developed by some researchers. We capture the issues, corresponding components, and future direction of those methods. Furthermore, we propose a smart contract development model based on our findings. This paper's structure is as follows: Section 2 provides an overview of blockchain technology, decentralized applications, and smart contracts. Section 3 sets out the selected research methodology, presents the search process's workflow, identifies the screening technique, analyzes/classifies the selected resources, and displays the mapping process. Section 4 presents the results and answers to our research questions. Section 5 discusses the proposed novel smart contract development model. Section 6 shall conclude the paper.

2 BACKGROUND

2.1 Blockchain

Blockchain works as a distributed ledger that stores all the transaction data that occurs in the blockchain network. The data integrity is formed through a shared agreement between all the parties involved in the system, which is called a consensus mechanism [9]. In a public blockchain, the dependency on a central authority such as a bank or any financial institution is minimized as the transaction is validated through the network participants using that consensus mechanism. Technically, all the transaction verified by the participants in the blockchain network is stored in a block protected by cryptographic hash function [17]. The first block in a blockchain network is called the "genesis block" [6]. Figure 1 shows the generic structure of a Blockchain with its attributes in the block header.

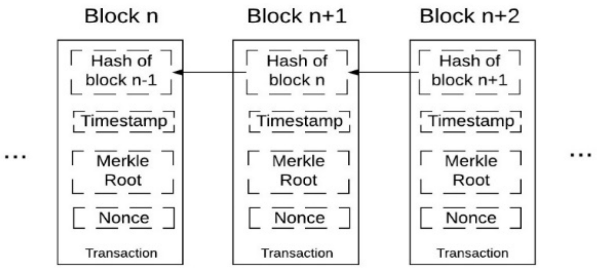

Figure 1: Generic Blockchain Structure.

2.2 Decentralized Applications and Smart Contract

A smart contract is a utility released on Ethereum Improvement Project (EIP) [12] in 2015. The smart contract allows the user to create a self-executed algorithm with immutable rules on the blockchain system. With smart contracts, users can create an agreement that rules all the transactions in the application that formed on the Ethereum platform. The applications are built on the Ethereum platform and use a smart contract that ordering the transaction is called Decentralized Applications (Dapps). Figure 2 shows the architecture of Decentralized Application in Ethereum platform.

The development of smart contracts in the Ethereum platform is using the Solidity programming language. The smart contract is compiled and run on the top of a gigantic and distributed virtual machine called Ethereum Virtual Machine (EVM). In the typical Dapps architecture, a smart contract works in the application layer that runs the algorithms and application rule. Code efficiency in smart contract development is critical as the smart contract's programming logic and storage is related to the gas cost required to execute the program [23]. Besides the smart contract, Dapps also need an off-chain application such as a database layer, application layer, and front-end application. The development of Dapps could be challenging related to the integration of all the components above.

3 RESEARCH METHOD

The systematic mapping method was used to investigate studies relevant to the smart contract development process method in blockchain technology. The results of this method's application have helped the authors identify and map the related papers and articles that will enable us to answer more research questions. A systematic mapping method aims to define the state of understanding of issues or topics [25]. Systematic mapping gathers, discusses,

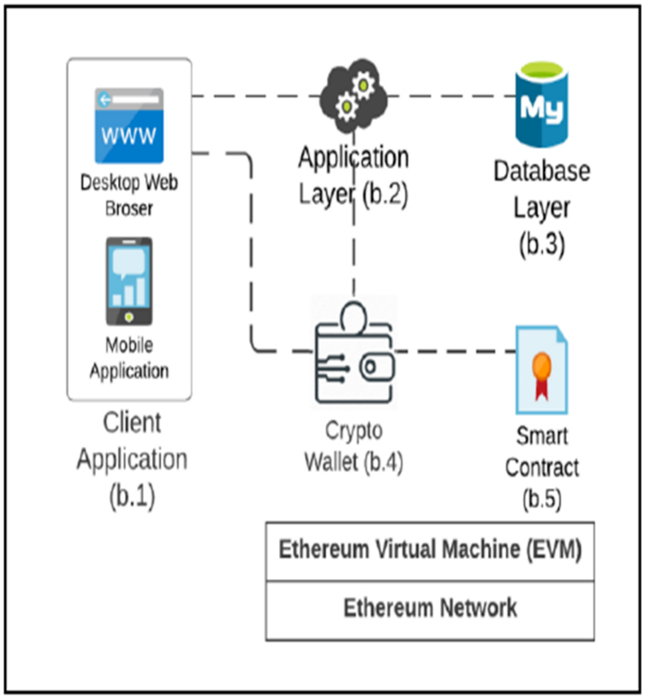

Figure 2: Decentralized Application in Ethereum.

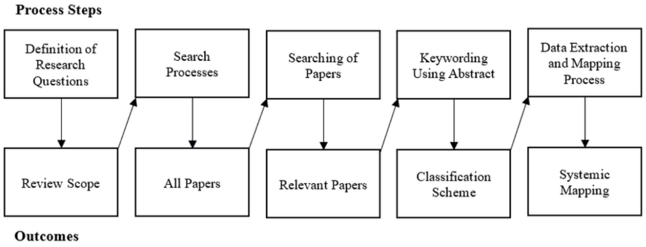

Figure 3: Systematic Mapping Process (Petersen, 2008).

and catalogs the evidence available on the topic of interest, which, in our case, is the Smart Contract development process. Systematic mapping is an innovative method for collecting evidence in environmental science. As a result of its broad relevance and usability, it has become a standard form of evidence synthesis.

Figure 3 displays the mapping method workflow and all the steps we have gone through in this study. The method consists of five steps, each with a corresponding outcome.

3.1 RQs Definition

Our study’s overall objective is to gain an insight into the problems that relate to the smart contract development process and their corresponding solutions. To obtain a detailed overview of this subject area, we address three research questions (RQs). Table 1 presents our three RQs along with their motivations.

3.2 Search Process

The search took place between June 2020 and October 2020. We used several search engines/databases to search the articles: Google Scholar, Springer Link, Science Direct, and IEEE Xplore. We used various keywords (e.g., Smart Contracts, Method, Development, Framework, and Decentralized Applications) that apply to our research topic and made various combinations between them in the search engines listed above. A collection of 475 articles was the result of this.

3.3 Searching of Papers

After the queries were made, the papers were screened to find the relevant ones by examining the title, abstract, findings, conclusions, number of citations, and publication year. Table 2 displays the inclusion and exclusion criteria used to exclude papers that did not match this study. The following two filters were applied in order to address the research questions:

Filter 1: Returned papers were scanned by reading the title, keywords, summary, findings, conclusion, number of citations, and year of publication, and by applying the inclusion/exclusion criteria in Table 2

3.4 Keywording

The data extraction strategy was providing and sorting the necessary sets of data for answering each RQ.

- **RQ1**: To identify the critical issues related to the smart contract development, we focused on the background and discussion section and identified the issue related to the smart contract development process.
- **RQ2**: We study the entire context of the paper and determine the stakeholders that are related to the paper.
- **RQ3**: To Identify the future development process, we need to first extract the issues and stakeholders from RQ1 and RQ2. Then we propose a new smart contract development process that satisfied our review result

3.5 Data Extraction and Mapping Process

Here, we gathered all the information needed to answer the research questions. After collecting and presenting data in tables, we also plotted graphs that would allow us to conclude. These data elements are the key goals and achievements of the selected articles.

4 RESULT

4.1 Selection Result

Out of 475 papers returned, 105 papers left when filtering in Section 3.3 was applied. The remaining 22 papers were extensively examined and analyzed to react to the RQs in Table 1. Figure 4 shows the publication years of the selected papers.

4.2 RQs Result

RQ1: WHAT ARE THE ISSUES RELATED TO SMART CONTRACT DEVELOPMENT PROCESS?

After reading our paper source entirely, we found that security related to the smart contract vulnerability is the most mentioned

Table 1: Research Questions

No	Research Question	Motivation
RQ 1	What are the issues related to the smart contract development process?	To know the issues behind the proposed smart contract development.
RQ 2	What are the corresponding components defined in the smart contract development process?	Determine components that are defined in the smart contract development process.
RQ 3	How should the smart contract development process be considered in the future?	The needs of the smart contract development process that covers several issues related to smart contract development

Table 2: Inclusion and Exclusion Criteria

Inclusion Criteria	Exclusion Criteria
The paper is in the area of Blockchain and Smart Contract	The paper is not related to the smart contract development process
The full text of the paper is available	The full text of the paper is not available
The paper presents a Smart Contract Development	The experiment is not complete
The paper has a good quality of research	

Filter 2: The papers that passed Filter 1 have been read in their entirety and summarized in Table 3

Table 3: Inclusion and Exclusion Criteria

Returned Papers	Filter 1	Filter 2
475	105	22

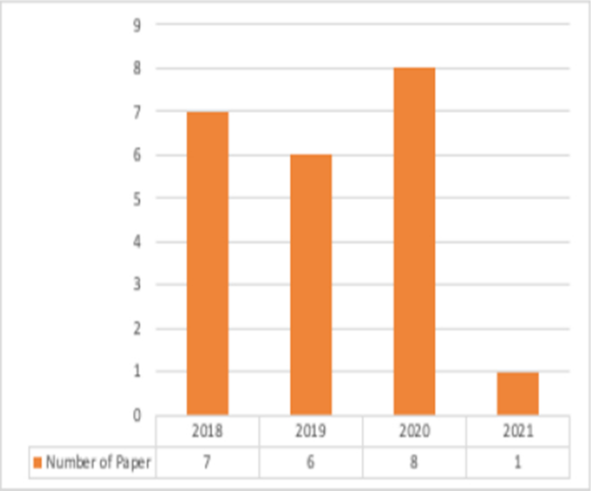

Figure 4: Selection Result.

issue in the paper [1, 2, 18, 21, 22, 29, 32, 33, 37, 38]. The smart contract vulnerability, including Transaction-Ordering Dependence, Timestamp Dependence, Mishandled Exceptions, Re-entrancy, and Callstack Depth [1, 32]. Besides the contract vulnerability, the blockchain platform's limitation, including Irreversible Bugs, Performance Issues, Lack of Oracles and Standards, also considered important issues in the smart contract development process. Furthermore, we identify another blockchain limitation issue, such as uncertainty, unification, cross-chain capabilities, and extensibility [13, 16, 36].

Another exciting issue we found in this study is contract abstraction [5, 15, 22, 35]. Smart contract development and programming task is deterministic and low-level activity. The purpose of smart contract abstraction is to fill the gap between programming and business logic. This issue subsequently leads us to another issue related to a smart contract's time and complexity [4, 5] . From this point, the complexity of smart contract development is also facing significant issues related to the need for a standardized development method that ready for further development complexity [7, 19, 27, 31].

- RQ2: WHAT ARE THE CORRESPONDING COMPONENTS DEFINED IN THE SMART CONTRACT DEVELOPMENT PROCESS?

The security mechanism becomes the most mentioned component in our study. The smart contract development process needs to use a security mechanism to address contract vulnerability issues mentioned in RQ1 [1, 33]. There are several security components needed for the smart contract development process, such as input filter [34], verification tool [21], and security checklist [18]. From the perspective of code development, library construction, and resource management is needed to ensure the code development running smoothly with the uncertainty of blockchain technology in the future [38]. Language and Tech Stack selection also become essential components needed in the code development process [4, 15]. Further, the smart contract development process also involves high-level components such as requirement analysis [7], design [16, 27], modeling [5, 22, 35], and testing [31]. This finding shows that the smart contract development process could be accomplished using a common software development method that emphasizes the security issue.

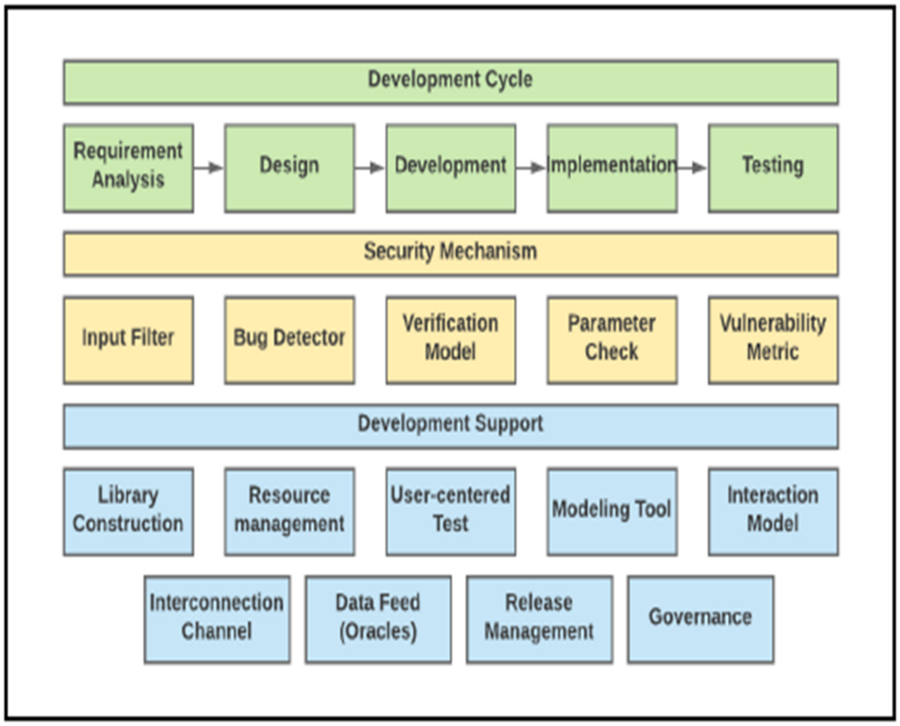

Figure 5: Smart Contract Development Model.

- RQ3: HOW SMART CONTRACT DEVELOPMENT PROCESS SHOULD BE CONSIDERED IN THE FUTURE?

Related to the previous RQs, the smart contract is very exposed to the security threat. The contract vulnerability leads the decentralized application to very high risk. Hence, the development process of a smart contract should seriously address these issues. Our findings in RQ2 shows that the smart contract development process could be accomplished by using a common software development method but need more focus on the security issue.

5 DISCUSSION

From the identified result from our RQs, we propose a model for smart contract development. This model is divided into three blocks interconnected, including Development Cycle, Security Mechanism, and Development Support. Each block has a different function in exercise the smart contract development process, which is the goal of this model. The blocks are distinguished with three colors (green, yellow, and blue) and designed to complete another block's function. Figure 5 shows our proposed model from this study.

Following are the elaboration of each block:

- 1. Development Cycle

The development cycle is adopted from the common software development process. Requirement analysis, design, development, implementation, and testing are subsequentially needed in a smart contract development process. The interconnection and support from the other blocks will create advantages for each process inside this cycle.

- 2. Security Mechanism

The security mechanism in our model is designed by considering our findings in this study. The input filter and bug detector are beneficial to conduct early security checks in the smart contract development process. The verification model and parameter check can be used along with the smart contract development process to assess every contract's interaction. Last, the vulnerability metric is used to discover and assess the vulnerability of every smart contract.

- 3. Development Support

This block is designed to support all the activities in the development cycle. The activity is including library construction, resource management, user-centered test, modeling tools, interaction model, interconnection channel, data feed (oracles), release management, and governance—our findings in this study design these activities. Every activity could be used by tailoring the activity to the development cycle. Library construction, resource management, modeling tools, data feed, and release management could be tailored to the development cycle's code development process. Meanwhile, the user-centered test activity could be used in the testing and requirement analysis process.

6 CONCLUSION

This study is driven by the uncertainty and needs of the standard in creating a smart contract. The uncertainty, complexity, and time needed in the development process of a smart contract could potentially obstruct blockchain adoption in the future. We conducted a study that founded several issues and components that were needed in the development process. We find that security is an essential issue in smart contract development, especially related to contract vulnerability. This vulnerability will lead us to the need for a proper code development that should be designed to address the exposure of contract vulnerability itself.

Therefore, our proposed solution addresses all the activities needed in the smart contract development process and focuses on security threats and exposure. This model could lead to a secure-oriented smart contract development process in creating a decentralized application. For future study, we recommend the other researcher create an experiment related to the decentralized application using our proposed model. We also encourage this model to be used as evaluation tools to assess the feasibility of a smart contract.

REFERENCES

[1] Ayman, A. *et al.* 2020. Smart Contract Development from the Perspective of Developers: Topics and Issues Discussed on Social Media. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). R. Böhme *et al.*, eds. Springer Berlin Heidelberg. 405–422.

[2] Bai, X. *et al.* 2018. Formal Modeling and Verification of Smart Contracts. Proceedings of the 2018 7th International Conference on Software and Computer Applications - ICSCA 2018 (New York, New York, USA, 2018), 322–326.

[3] Buterin, V. 2014. A next-generation smart contract and decentralized application platform. Etherum. January (2014), 1–36.

[4] Cheshun, V. *et al.* 2020. Safe Decentralized Applications Development Using Blockchain Technologies. 2020 10th International Conference on Advanced Computer Information Technologies (ACIT) (Sep. 2020), 800–805.

[5] Choudhury, O. *et al.* 2018. Auto-Generation of Smart Contracts from Domain-Specific Ontologies and Semantic Rules. 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (Jul. 2018), 963–970.

[6] Christidis, K. and Devetsikiotis, M. 2016. Blockchains and Smart Contracts for the Internet of Things. IEEE Access. 4, (2016), 2292–2303. DOI:https://doi.org/10.1109/ACCESS.2016.2566339.

[7] Coblenz, M. *et al.* 2019. Smarter Smart Contract Development Tools. 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB) (May 2019), 48–51.

[8] COLLOMB, A. *et al.* 2019. Blockchain Technology and Financial Regulation: A Risk-Based Approach to the Regulation of ICOs. European Journal of Risk Regulation. 10, 2 (Jun. 2019), 263–314. DOI:https://doi.org/10.1017/err.2019.41.

[9] Conoscenti, M. *et al.* 2016. Blockchain for the Internet of Things: A systematic literature review. 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA) (Nov. 2016), 1–6.

[10] Dannen, C. 2017. Bridging the Blockchain Knowledge Gap. Introducing Ethereum and Solidity. Apress. 1–20.

[11] Dowlat, S. 2018. Cryptoasset Market Coverage Initiation: Network Creation.

[12] EIP 20: ERC-20 Token Standard: 2015. https://eips.ethereum.org/EIPS/eip-20. Accessed: 2020-04-30.

[13] Falazi, G. *et al.* 2019. Modeling and execution of blockchain-aware business processes. SICS Software-Intensive Cyber-Physical Systems. 34, 2–3 (Jun. 2019), 105–116. DOI:https://doi.org/10.1007/s00450-019-00399-5.

[14] Hackers just tapped China's dForce for $25 million in Ethereum exploit - Decrypt: 2020. https://decrypt.co/26033/dforce-lendfme-defi-hack-25m. Accessed: 2020-07-14.

[15] He, X. *et al.* 2018. SPESC: A Specification Language for Smart Contracts. Proceedings - International Computer Software and Applications Conference. 1, (2018), 132–137. DOI:https://doi.org/10.1109/COMPSAC.2018.00025.

[16] Karamitsos, I. *et al.* 2018. Design of the Blockchain Smart Contract: A Use Case for Real Estate. Journal of Information Security. 09, 03 (2018), 177–190. DOI:https://doi.org/10.4236/jis.2018.93013.

[17] Lee, W.-M. 2019. Beginning Ethereum Smart Contracts Programming. Apress.

[18] Marchesi, L. *et al.* 2020. Security checklists for Ethereum smart contract development: patterns and best practices. (Aug. 2020), 1–13.

[19] Marchesi, M. *et al.* 2018. An Agile Software Engineering Method to Design Blockchain Applications. Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia on ZZZ - CEE-SECR '18 (New York, New York, USA, 2018), 1–8.

[20] Marino, B. and Juels, A. 2016. Setting standards for altering and undoing smart contracts. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 9718, (2016), 151–166. DOI:https://doi.org/10.1007/978-3-319-42019-6_10.

[21] Mavridou, A. *et al.* 2019. VeriSolid: Correct-by-Design Smart Contracts for Ethereum. Financial Cryptography and Data Security. Lecture Notes in Computer Science. 446–465.

[22] Mavridou, A. and Laszka, A. 2018. Designing Secure Ethereum Smart Contracts: A Finite State Machine Based Approach. 523–540.

[23] McCorry, P. *et al.* 2017. A Smart Contract for Boardroom Voting with Maximum Voter Privacy. Financial Cryptography and Data Security. FC 2017. Lecture Notes in Computer Science. 357–375.

[24] Parizi, R.M. *et al.* 2018. Smart Contract Programming Languages on Blockchains: An Empirical Evaluation of Usability and Security. Proceedings Blockchain-ICBC. Springer International Publishing. 75–91.

[25] Petersen, K. *et al.* 2008. Systematic Mapping Studies in Software Engineering. International Journal of Software Engineering & Knowledge Engineering (Jun. 2008), 33–55.

[26] Ranganthan, V.P. *et al.* 2018. A decentralized marketplace application on the ethereum blockchain. Proceedings - 4th IEEE International Conference on Collaboration and Internet Computing, CIC 2018. (2018), 90–97. DOI:https://doi.org/10.1109/CIC.2018.00023.

[27] Sillaber, C. *et al.* 2020. Laying the foundation for smart contract development: an integrated engineering process model. Information Systems and e-Business Management. 0123456789 (Feb. 2020). DOI:https://doi.org/10.1007/s10257-020-00465-5.

[28] Sillaber, C. and Waltl, B. 2017. Life Cycle of Smart Contracts in Blockchain Ecosystems. Datenschutz und Datensicherheit - DuD. 41, 8 (2017), 497–500. DOI:https://doi.org/10.1007/s11623-017-0819-7.

[29] Syahputra, H. and Weigand, H. 2019. The Development of Smart Contracts for Heterogeneous Blockchains. Proceedings of the I-ESA Conferences. Springer International Publishing. 229–238.

[30] Szabo, N. 1996. Smart contracts: building blocks for digital markets. EXTROPY: The Journal of Transhumanist Thought, (16). 18, (1996), 2.

[31] Vilain, P. *et al.* 2020. A preliminary study on using acceptance tests for representing business requirements of smart contracts. IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2020. (2020). DOI:https://doi.org/10.1109/ICBC48266.2020.9169480.

[32] Wang, S. *et al.* 2019. Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 49, 11 (2019), 2266–2277. DOI:https://doi.org/10.1109/TSMC.2019.2895123.

[33] Wang, Z. *et al.* 2021. Ethereum smart contract security research: survey and future research opportunities. Frontiers of Computer Science. 15, 2 (Apr. 2021), 152802. DOI:https://doi.org/10.1007/s11704-020-9284-9.

[34] Wang, Z. *et al.* 2020. FSFC: An input filter-based secure framework for smart contract. Journal of Network and Computer Applications. 154, June 2019 (Mar. 2020), 102530. DOI:https://doi.org/10.1016/j.jnca.2020.102530.

[35] Wohrer, M. and Zdun, U. 2020. Domain Specific Language for Smart Contract Development. 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (May 2020), 1–9.

[36] Wu, X. *et al.* 2020. ChainIDE 2.0: Facilitating smart contract development for consortium blockchain. IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2020. (2020), 388–393. DOI:https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9163051.

[37] Yuan, R. *et al.* 2018. ShadowEth: Private Smart Contract on Public Blockchain. Journal of Computer Science and Technology. 33, 3 (May 2018), 542–556. DOI:https://doi.org/10.1007/s11390-018-1839-y.

[38] Zou, W. *et al.* 2019. Smart Contract Development: Challenges and Opportunities. IEEE Transactions on Software Engineering. PP, March 2018 (2019), 1–1. DOI:https://doi.org/10.1109/TSE.2019.2942301.