

Smart Contract Development from the Perspective of Developers: Topics and Issues Discussed on Social Media

Afiya Ayman, Shanto Roy, Amin Alipour, and Aron Laszka

University of Houston

Published in the proceedings of the 4th Workshop on Trusted Smart Contracts (WTSC) in association with Financial Cryptography (FC 2020).

Abstract. Blockchain-based platforms are emerging as a transformative technology that can provide reliability, integrity, and auditability without trusted entities. One of the key features of these platforms is the trustworthy decentralized execution of general-purpose computation in the form of smart contracts, which are envisioned to have a wide range of applications. As a result, a rapidly growing and active community of smart-contract developers has emerged in recent years. A number of research efforts have investigated the technological challenges that these developers face, introducing a variety of tools, languages, and frameworks for smart-contract development, focusing on security. However, relatively little is known about the community itself, about the developers, and about the issues that they face and discuss. To address this gap, we study smart-contract developers and their discussions on two social media sites, Stack Exchange and Medium. We provide insight into the trends and key topics of these discussions, into the developers' interest in various security issues and security tools, and into the developers' technological background.

1 Introduction

The popularity and adoption of blockchain based platforms are growing rapidly both in academia and industry. This growth is driven by the unique features of blockchains: providing integrity and auditability for transactions in open, decentralized systems. While earlier blockchains, such as Bitcoin [42], used these features to establish cryptocurrencies, more recent blockchains, such as Ethereum, also function as distributed computational platforms [56,61]. These platforms enable developers to deploy general-purpose computational code in the form of smart contracts, which can then be executed by a decentralized but trustworthy system. Smart contracts are envisioned to have a range of innovative applications, such as asset tracking in the Internet of Things [10], privacy-preserving transactive energy systems [31,59], and various financial applications [51].

Since the security of these applications hinges on the correctness of the underlying contracts, it is crucial that developers are able to create correct contracts. Sadly, the development of smart contracts has proven to be a challenging and error-prone process, in large part due to the unusual semantics of smart contract platforms and languages [4,36]. Studies have found that a large number of contracts that are deployed on the main Ethereum network suffer from various security issues [36,44]. Such issues may manifest as security vulnerabilities, some of which have led to security incidents with financial losses in the range of hundreds of millions of dollars worth of cryptocurrencies [20,43]. As a response, the research community has stepped forward and introduced a number of tools (e.g., [2,36,55,44]), frameworks (e.g., [37,39,38]), and even new languages (e.g., [45]) to help developers.

While the technical capabilities of these tools and frameworks have been evaluated by multiple surveys (e.g., [13,29,9,35,23]), relatively little is known about whether developers use them in practice or even whether developers are aware of them. In fact, to the best of our knowledge, no prior work has studied the smart contract developers’ awareness of security issues and tools or about which issues they are most concerned. In light of this, there is a clear gap in research regarding the developers’ perspective of smart contract development. Further, very little is known about the developers’ technological background and interests, and about their online communities. Such information is crucial for enabling researchers to better understand the potential entry barriers for smart contract technology and for guiding researchers to address the developers’ needs.

To address this gap, we study the smart contract developers’ online communities, the topics that they discuss, and their interest in various security issues and tools. To this end, we collect data from three social media sites: *Stack Overflow*, the most popular Q&A site for software developers [6,54,3]; *Ethereum Stack Exchange*, a site focusing on Ethereum from the leading network of Q&A sites [17], and *Medium*, a popular blog hosting site [11]. In particular, we collect and analyze discussions about smart contracts (e.g., posted questions, answers, blog entries, comments) as well as information about the users who participate in these discussions. We seek to answer the following research questions:

- Q1 Trends:** What are the main trends in smart contract related discussions? How do they compare to discussions related to other technologies?
- Q2 Security:** Which common security issues and tools do developers discuss? Do discussions about security issues and tools coincide?
- Q3 Developers:** What are the smart contract developers’ technological background and interests besides smart contracts?

We answer the above questions in our analysis (Section 3); here, we highlight a few interesting results. We find that the intensity of smart contract related discussions reached its peak in early 2018 and has been slowly declining since then (while discussions about other technologies have remained stable). This coincides with the decline of ETH price, which peaked in January 2018. In the terminology of the so-called ‘hype cycle’ [19], this suggests that smart contracts may have passed the ‘peak of inflated expectations’ and are now in the ‘trough of

disillusionment’ phase. This is in interesting contrast with a 2019 July Gartner report [33], which placed smart contracts at the peak of expectations. On Stack Overflow and Ethereum Stack Exchange, we find that most questions about smart contracts receive at least one answer, while the majority of questions about other technologies remain unanswered; however, questions about smart contracts are less likely to lead to lengthy discussions. We also find that very few discussions are related to security, with re-entrancy being the most discussed vulnerability (in part due to the so-called “DAO attack”, as evidenced by our findings). There are even fewer mentions of security tools, even in security related discussions. However, we find a significantly higher number of security related posts on Medium. On all sites, smart contract related discussions are dominated by a few key technologies and languages (e.g., Solidity, web3.js, Truffle). Besides smart contracts, the topics that are most discussed by smart contract developers on Stack Overflow are related to web (e.g., jQuery, HTML, CSS). On Medium, we find that smart contract developers are also interested in non-technical topics related to entrepreneurship (e.g., startups, finance, investing).

Outline The remainder of this paper is organized as follows: In Section 2, we describe our data collection and analysis methodology. In Section 3, we present the results of our study. In Section 4, we give a brief overview of related work. Finally, in Section 5, we discuss our findings and provide concluding remarks.

2 Study Design

2.1 Data Collection

Research Ethics Our study is based on publicly available data, and we report statistical results that contain no personally identifiable information.

Stack Exchange is a network of question-and-answer (Q&A) websites. We collect data from two Stack Exchange sites: *Stack Overflow*¹, the most popular generic site for developers [6,54,3], and *Ethereum Stack Exchange*², the site that focuses on Ethereum. On these two websites, posts have the same structure: each post includes a question, a title, a set of associated tags, a set of answers, and a set of comments. Only registered users can post new questions or answer existing ones, which enables us to study the developers. To facilitate searching and categorizing posts, Stack Exchange requires users to associate one or more tags with each question. These tags are unstructured and chosen by the users, so they include a wide range of terms (e.g., *Python*, *linked-list*).

From Ethereum Stack Exchange, we collect all posts and users using Stack Exchange Data Explorer (SEDE) [1]. We also collect all posts and users from Stack Overflow using the quarterly archives hosted on the Internet Archive [24], which we complement with the latest data from SEDE. Since Stack Overflow

¹ <https://stackoverflow.com/> ² <https://ethereum.stackexchange.com/>

is a generic site for developers, we need to find posts that are related to smart contracts. To this end, we use a snowballing methodology. First, we find all posts whose tags contain *smartcontract*. Then, we extract other tags from the collected posts, identify the most frequently used tags that are strictly related to smart contracts, and extend our search with these tags. We continue repeating this process until we cannot collect any more related posts. In the end, we search for posts whose tags contain the following strings (except for *ether*, which needs to be an exact match to avoid finding, e.g., *ethernet*): *smartcontract*, *solidity*, *ether*, *ethereum*, *truffle*, *web3*, *etherscan*. Finally, we manually check a random sample of the collected posts to confirm that they are indeed related to smart contracts. In total, we collect 30,761 smart contract related questions, 38,152 answers, and 73,608 comments as well as the 56,456 users who posted these. Our dataset includes everything up to November 22, 2019.

Medium³ is a popular blog platform [11], where registered users can publish posts on a variety of subjects, and other users may read, respond (i.e., comment), clap, or vote. A Medium post typically contains a title, a text body, tags, reader responses, number of claps and votes, author’s name and profile URL, reading time based on word count, and publication date. Since Medium is a generic blog site, we again use a snowballing methodology to collect smart contract related posts, similar to Stack Overflow. We first search for posts that contain the tag *smart contract*, and then iteratively extend our search with new tags, finally stopping at the following list of tags: *solidity*, *smart contract*, *smart contracts*, *viper*, *metamask*, *truffle*, *erc20*, *web3*. Again, we manually check a random sample to confirm that the collected post are indeed related to smart contracts. In total, we collect 4,045 unique posts from 2,165 authors, which have been posted on Medium between January 2014 and November 24, 2019.

2.2 Methodology

Statistical Analysis First, we analyze various statistics of smart contract related posts and the posting users from Stack Exchange and Medium. Statistics for posts include the rate of new posts over time, the distributions of tags, number of answers, etc., while statistics for users include the distribution of tags in all of their posts. For the Stack Exchange dataset, we also compare smart contract related posts to posts about other subjects on Stack Overflow.

Textual Data Analysis Next, we preprocess the data to prepare the posts for text analysis. First, for each Stack Exchange post, we combine the title, question, answers, comments, and tags together; for each Medium post, we combine the title, text, and tags together. Second, we remove HTML tags and code snippets from all posts. After this step, we search for occurrences of certain keywords in the posts, such as mentions of common security issues and tools.

³ <https://medium.com/>

3 Results

3.1 Discussion Trends (Q1)

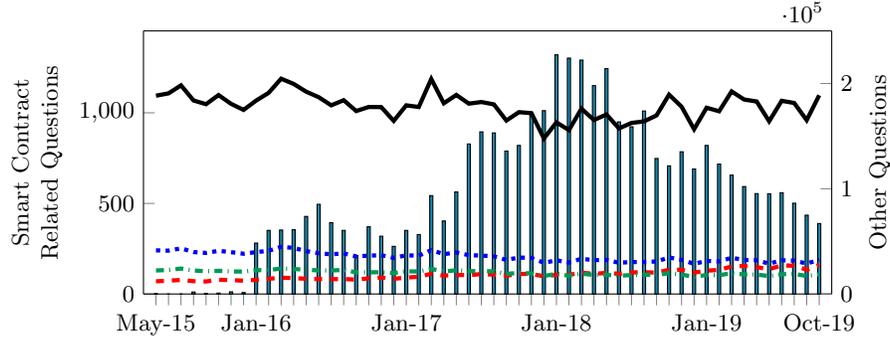


Fig. 1: Number of smart contract related questions (vertical bars) posted on Stack Exchange, number of all questions (black line) and Java (dotted blue), Python (dashed red), and JavaScript (dash-dotted green) related questions posted on Stack Overflow each month. Please note the different scales.

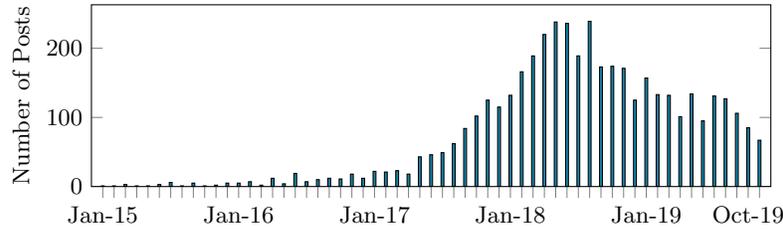


Fig. 2: Number of smart contract related posts on Medium each month.

We begin our analysis by comparing trends in posts about smart contracts with trends in posts about other technologies (e.g., Java and Python). Specifically, we study how interest in smart contracts (measured as the number of new posts) has evolved over time and how active smart contract related discussions are (measured using distributions of answers and comments), showing significant differences compared to other technologies.

Figure 1 compares the number of questions related to smart contracts (vertical bars) posted on Stack Exchange with the total number of questions (black line) posted on Stack Overflow each month. For the sake of comparison, the figure also shows numbers of questions about other, more mature technologies, namely Java (dotted blue), Python (dashed red), and JavaScript (dash-dotted

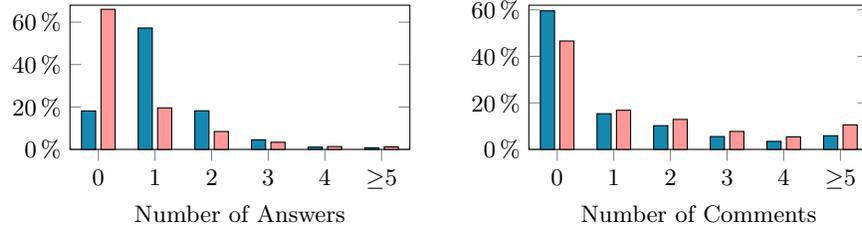


Fig. 3: Number of answers and comments received by smart contract related questions (blue) and by other questions (red) on Stack Exchange.

green). The first smart contract related questions were posted in May 2015, but users did not start posting in significant numbers until Ethereum Stack Exchange launched in January 2016. From 2017 to early 2018, there is a clear upward trend; however, the rate of new questions has been steadily declining since then, which suggests that interest in smart contracts on Stack Exchange peaked in early 2018. Meanwhile, the overall rate of new questions on Stack Overflow has remained steady since 2015. Similarly, the rates of new questions about Java, Python, and JavaScript have remained relatively steady, with only slightly increasing (Python) and decreasing (Java) trends. These results suggest that the significant fluctuations observed in smart contract related questions are not due to the varying popularity of Stack Overflow. Finally, Figure 2 shows the number of new Medium posts related to smart contracts in each month. Again, we observe a clear upward trend from 2017 to early 2018, peaking in the first half of 2018, and a steady decrease since then.

The close similarity between Figures 1 and 2 suggests that our observations are robust in the sense that they are not artifacts of our data sources, our measurement, or our analysis; rather, the trends that we observe may be signs of declining developer interest in smart contracts. Further, this finding is corroborated by a third data source, the price of Ethereum (ETH). Indeed, we can see a very similar trend in the price of ETH⁴ over the past years: ETH reached its highest value on January 12, 2018 [16] and has been mostly declining since then.

To gain insight into the level of interactions in the smart contract developer community, we analyze the distributions of answers and comments in smart contract related posts. Figure 3 shows the number of answers and comments received by smart contract related questions (blue) on Ethereum Stack Exchange and Stack Overflow and by other questions (red) on Stack Overflow. We observe that 82% of smart contract related questions have at least one answer. This ratio is very high compared to other questions, of which less than 34% have at least one answer. We speculate that this difference could be explained by smart contract related questions being simpler and easier to answer, asking mostly about basic issues; or it could be explained by the community of smart contract developers being more active. However, we also observe that few smart

⁴ www.coinbase.com/price/ethereum

Table 1: Most Frequent Tags in Smart Contract Related Posts

Stack Exchange						Medium				
Tag	Num.	Average				Tag	Num.	Average		
		Score	View	Ans.	Com.			Resp.	Clap	Voter
Solidity	9323	0.48	752	1.2	1.14	Ethereum	2643	2.37	388	37.10
Go-Ethereum	4946	0.55	1047	1.09	1.19	Blockchain	2585	2.06	423	35.91
web3js	3948	0.48	880	1.16	1.37	Smart Contracts	1274	1.68	311	32.03
Contract-development	2973	0.70	845	1.29	1.04	Solidity	907	1.62	290	29.26
Blockchain	2539	0.88	1232	1.53	1.37	Cryptocurrency	659	2.48	577	41.32
Ethereum	2530	1.55	3023	3.73	3.94	Security	476	0.81	194	16.50
Truffle	2430	0.40	750	1.28	1.46	ERC20	467	3.63	836	54.46
Transactions	1743	0.94	1382	1.31	1.1	Web3	401	1.75	429	41.03
Remix	1642	0.29	593	1.15	1.34	Bitcoin	369	5.04	730	74.07
Contract-design	1522	1.12	873	1.34	0.92	MetaMask	296	0.76	216	16.97

contract related questions receive more than one answer, and very few receive five or more, especially in comparison with other questions. This suggests that the more likely explanation is that smart contract related questions are indeed simpler since developers rarely post improved or conflicting answers. We also observe that smart contract related questions tend to receive fewer comments than other questions, and receiving five or more comments is very rare (5.9% of smart-contract questions vs. 10.6% of other questions). While the difference compared to other questions is not as pronounced as in the case of answers, this difference indicates that smart contract related questions rarely spark lengthy debates or discussions. This might again suggest that questions pertain to simpler issues.

Finally, we study what topics are at the center of smart contract related discussions. Posts from both Stack Exchange and Medium have tags to identify the topic of discussion. Although these tags do not necessarily capture the exact topic of discussion, they can indicate what technologies, issues, etc. are discussed. Table 1 lists ten tags that are most frequently used in smart contract related posts on each site. For Stack Exchange, the table lists the average score⁵ and the average number of views, answers, and comments received by questions with each tag. For Medium, it lists the average number of responses, claps, and number of voters for each tag. The list is dominated by a few smart contract technologies, such as *Solidity* (high-level language for smart contracts), *Go-Ethereum* (official implementation of Ethereum), *web3js* (JavaScript library for Ethereum), and *Truffle* (development environment for smart contracts). We found very few mentions of other smart-contract languages, such as Vyper or Plutus [57,25]. This may suggest the existence of a monoculture: most develop-

⁵ Score is the difference between the number of upvotes and downvotes for a post.

ers might be familiar with only a small set of technologies. Alternatively, it may indicate that some technologies are discussed mostly on alternative forums.

3.2 Security Issues and Tools (Q2)

Next, we focus on discussions related to security. Our goal is to gauge the smart contract developers’ level of concern and awareness about various security issues and tools. To this end, we search for posts related to common security issues and tools, using the numbers of related posts as indicators for concern about security and for awareness about tools.

Table 2: Posts Mentioning Common Security Issues

Security Issues	Stack Exchange		Medium	
	Number	Percentage	Number	Percentage
Re-Entrancy	126	0.41%	164	4.05%
Denial of Service	95	0.31%	111	2.74%
Race Condition	35	0.11%	34	0.84%
Integer Overflow	16	0.05%	95	2.35%
Transaction-Ordering Dependence	4	0.01%	66	1.63%
Timestamp Dependence	4	0.01%	49	1.21%
Integer Underflow	2	0.007%	12	0.30%

Security Issues To gauge how concerned smart contract developers are about security, we first search for mentions of *security* and *vulnerability* in smart contract related posts. We search in the preprocessed texts of the posts, which include tags, titles, comments, etc. by considering all common variations of these terms (e.g., *vulnerabilities* and *vulnerable*). On Stack Exchange, we find 1,211 and 236 posts that mention *security* and *vulnerability*, respectively, which constitute only 3.9% and 0.77% of all smart contract related posts. On Medium, we find 1,429 and 470 posts that mention *security* and *vulnerability*, respectively, which constitute 32% and 11% of all smart contract related posts. Based on these findings, we speculate that security awareness on Stack Exchange is rather low, while it is comparatively high on Medium. Unfortunately, many developers use Stack Exchange as their primary source of information [6].

Next, we consider specific types of vulnerabilities. Based on prior surveys of smart contract vulnerabilities [36,4,32,63,9], we establish the following list of common issues to search for: *re-entrancy*, *timestamp dependence*, *transaction-ordering dependence*, *integer overflow*, *integer underflow*, *race condition*, and *denial of service*. Again, we search for mentions of these issues in the preprocessed posts by considering all common variations (e.g., *DoS*, *dependence* and *dependency*). Table 2 shows the number of smart contract related Stack Exchange and

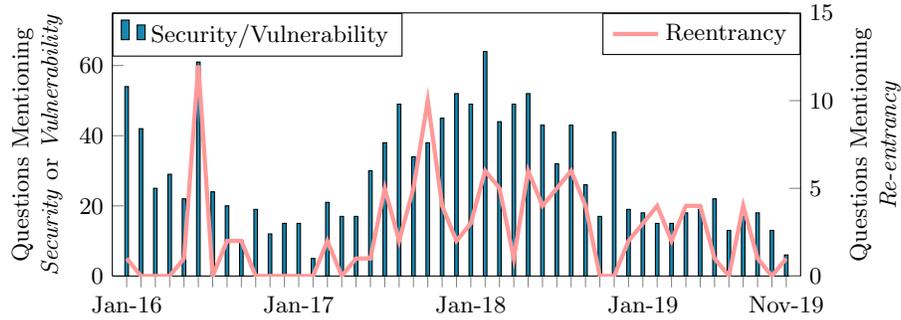


Fig. 4: Number of smart contract posts per month on Stack Exchange mentioning *security* or *vulnerability* and *re-entrancy*. Please note the different scales.

Medium posts that mention these issues. We find that Stack Exchange not only suffers from generally low security concern, but discussions are also restricted to only a few issues, such as *re-entrancy*; meanwhile, Medium posts discuss a broader range of issues. To explain Stack Exchange users’ fascination with the *re-entrancy* vulnerability, consider Figure 4, which shows the number of new posts mentioning *re-entrancy* for each month. There is a significant peak in 2016 June, which is when one of the most famous Ethereum security incidents happened, the so-called “DAO attack,” which exploited a *re-entrancy* vulnerability [20]. A significant number of security discussions on Stack Exchange seem to be driven by this incident. Also note that Figure 4 shows relatively high interest in security back in 2016. However, while the number of smart contract related posts on Stack Exchange rapidly rose in 2017, interest in security rather declined.

Security Tools, Frameworks, and Design Patterns We complement our results on security issues by studying the smart contract developers’ awareness of security tools (e.g., which tools they ask about or suggest in answers). We compile a comprehensive list of security tools based on relevant evaluation and survey papers (e.g., [13,29,9,35,23,47]) and other sources (e.g., [12]), and search for mentions of the following (in alphabetical order): ContractFuzzer [26], ContractLarva [15], echidna⁶, EtherTrust [21], EthIR, Ethlint (formerly known as Solium)⁷, FSolidM [37], MAIAN [44], Manticore [40], Mythril (as well as the service MythX and the client Mythos) [41], Octopus⁸, Osiris [53], Oyente [36], Rattle [50], ReGuard [34], SASC [62], sCompile [8], Securify [55], Slither [18], SmartAnvil [14], SmartCheck [52], solcheck⁹, solgraph¹⁰, solint¹¹, Solhint¹², SonarSolidity¹³, Sūrya (also spelled as Surya)¹⁴, teEther [30], Vandal [7], VeriSolid [39], VerX [48], VULTRON [58], Zeus [28]. Note that our goal is not to evaluate

⁶ github.com/crytic/echidna

⁸ github.com/quoscient/octopus

¹⁰ github.com/raineorshine/solgraph

¹² protofire.github.io/solhint

¹⁴ github.com/ConsenSys/surya

⁷ www.ethlint.com

⁹ github.com/federicobond/solcheck

¹¹ github.com/SilentCicero/solint

¹³ github.com/sagap/sonar-solidity

Table 3: Number of Posts Mentioning Various Security Tools and Patterns

Tools and Pattern	Stack Exchange	Medium	Tools and Pattern	Stack Exchange	Medium
Mythril	12	98	solcheck	2	5
Oyente	10	64	Maian	2	3
Smartcheck	4	57	Octopus	0	3
Securify	6	46	teEther	6	2
Solhint	8	39	Vandal	2	2
Ethlint/Solium	6	36	EthIR	2	2
scompile	0	33	SASC	1	2
Checks-Effects-Interactions	22	17	VeriSolid	0	2
Manticore	3	16	Zeus	1	1
Slither	0	10	Rattle	1	1
solgraph	2	7	ContractFuzzer	1	1
solint	2	6	SonarSolidity	3	0
Surya (Sūrya)	0	6	echidna	1	0

or compare the technical quality of these tools and frameworks (for that we refer the reader to surveys, e.g., [47]); we are only interested in whether they are discussed by developers. We also search for mentions of the *checks-effects-interactions* design pattern—considering again variations in spelling—which is meant to prevent the re-entrancy vulnerability [49].

Table 3 shows the number of smart contract related posts on Stack Exchange and Medium that mention the above tools. We again find low awareness on Stack Exchange: Mythril and Oyente are mentioned by only 12 and 10 posts, and other tools are mentioned by even fewer. However, we do find 22 posts that mention the *checks-effects-interactions* pattern, which is most likely due to interest in the *re-entrancy* vulnerability (see Table 2). Similarly, we again find higher awareness on Medium: there are 7 tools that are mentioned at least 33 times, with Mythril being mentioned the most.

Co-Occurrence of Security Issues and Tools Finally, we investigate if users recommend these tools against certain vulnerabilities and if they are aware of which vulnerabilities these tools address. To this end, we study which security issues and tools are mentioned together. Table 4 shows the number of posts on Stack Exchange and Medium that mention various pairs of security issues and tools (focusing on pairs mentioned by the most posts, omitting less frequent pairs). Again, we find low awareness on Stack Exchange: Mythril and Oyente are each mentioned only in 6 posts that also mention *security* or *vulnerability*, which means that these tools are suggested for security issues less than 0.5% of the time; other tools are mentioned even fewer times. These tools are not

Table 4: Co-Occurrence of Security Issues and Tools in Posts

Security Tools	Security /Vulnerability		Re-Entrancy		Timestamp Dependency		Transaction Ordering Dependency	
	Stack Overflow	Medium	Stack Overflow	Medium	Stack Overflow	Medium	Stack Overflow	Medium
Mythril	6	95	2	36	2	19	2	17
Oyente	6	63	2	37	1	26	1	3
Smartcheck	3	57	0	45	1	34	1	2
Securify	4	45	1	26	1	20	1	3
Solhint	2	38	1	34	1	26	1	1
Ethlint/Solium	4	28	1	10	1	6	1	3
Manticore	3	16	1	5	1	1	1	0
Slither	0	10	0	7	0	3	0	2
solgraph	2	7	1	4	1	1	1	0
Surya	0	6	0	2	0	1	0	0
solint	2	5	1	1	1	0	1	0
Solcheck	2	4	1	2	1	1	1	0
Maian	0	3	0	2	0	1	0	1
SASC	0	2	0	1	0	1	0	1
VeriSolid	0	2	0	1	0	0	0	0
Vandal	2	2	0	1	0	0	0	0
teEther	2	1	0	0	0	0	0	0
EthIR	2	1	0	1	0	0	0	0

mentioned even in conjunction with vulnerabilities that they address (see, e.g., *re-entrancy*). On the other hand, we find much higher awareness on Medium, as security issues and tools are often mentioned together.

3.3 Developers' Background and Interests (Q3)

For many developers, it is easier to adopt new tools, languages, and platforms that resemble ones with which they are already familiar. Hence, adoption of new technologies can hinge on the developers' technological background. To discover with which technologies smart contract developers are familiar, we study what tags they use in posts that are *not* related to smart contracts.

For each smart contract developer, we retrieve all of the developer's posts (i.e., questions and answers, or blog posts) that are *not* related to smart contracts, collecting a total of 1,250,325 posts from Stack Overflow and 44,684 posts from Medium. Table 5 lists the 10 most frequently used tags in the smart contract developers' Stack Overflow posts. The most frequent tags are all related to web development (*jQuery*, *HTML*, *CSS*, *Node.js*). Other popular tags correspond to major platforms (*.NET*, *Android*). Table 5 lists the most frequent tags from the smart contract developers' Medium posts in three categories: blockchain related, other technical, and non-technical (i.e., everything else). Note that since Medium is a generic blog site, there are many non-technical posts (e.g., tagged with *Business* or *Travel*). Unsurprisingly, the most popular tags are related to blockchains

Table 5: Other Tags used by Smart Contract Developers on SO & Medium

Stack Overflow		Medium					
		Blockchain		Technical (Other)		Non-Technical	
Tag	Freq.	Tag	Freq.	Tag	Freq.	Tag	Freq.
jQuery	8787	Blockchain	17877	Technology	2167	Startup	1897
HTML	6503	Cryptocurrency	8957	Artificial Intelligence	1331	Investing	958
CSS	5657	Bitcoin	5013	Fintech	1232	Finance	820
Node.js	5040	Crypto	2511	IoT	697	Business	786
.NET	4247	ICO	2256	Programming	646	Entrepreneurship	582
Android	3739	Security	630	JavaScript	635	Exchange	527
Objective-C	3727	Cryptocurrency Investment	620	Machine Learning	479	Marketing	493
MySQL	3330	Token Sale	616	Software Development	376	Innovation	488
Ruby	3281	Decentralization	525	Privacy	342	News	467
JSON	3231	Tokenization	220	Data	329	Travel	428

and cryptocurrencies. Other technical terms are led by the area of *Artificial Intelligence* and *Machine Learning*, and by tags related to software development (e.g., *Programming* and *JavaScript*). The most frequent non-technical terms are related to entrepreneurship (*Startup*, *Finance*, *Business*, *Investing*, etc.).

On both sites, we observe that a significant number of posts are related to JavaScript (highlighted in blue in Table 5): on Medium, *JavaScript* is the only programming language in the top 10 tags; on Stack Exchange, related technologies (*jQuery* and *Node.js*) are at the top. These results suggest that many smart contract developers have a background in JavaScript and related technologies, which may be explained by the similarity between JavaScript and Solidity, the most widely used high-level language for smart contract development.

4 Related Work

Smart Contract Development Practices Bartoletti et al. [5] were the first to quantitatively investigate the usage of design patterns and the major categories of smart contracts, providing a categorized and tabulated repository of data related to smart contracts. To this end, they examined smart contract platforms, applications, and design patterns, aggregated articles about smart contracts from [coindesk.com](#), and identified nine common design patterns used in some combination by most of the smart contracts that they found. Atzei et al. [4] presented a study of security vulnerabilities in Ethereum smart contracts, based on analysis of academic literature, Internet blogs, discussion forums about Ethereum, and practical experience in programming smart contracts. Based on their study, they provided a taxonomy for the root causes of vulnerabilities and techniques to mitigate them. Wohrer et al. [60] examined design patterns for smart contracts in Ethereum, focusing on two questions: which design patterns are common in

the ecosystem and how they map to Solidity coding practices. They employed a multivocal literature review, which considered various sources from academic papers to blogs and forums about Ethereum. Their analysis yielded 18 distinct design patterns. Jiang et al. [27] performed a preliminary study of blockchain technology as interpreted by developers and found that blockchain related questions represent a growing minority of posts on Stack Overflow. The most common problems with blockchain are related to configuration, deployment, and discussion, followed by ten less common categories. However, they did not consider the development of smart contracts.

Smart Contract Security Issues and Tools Parizi et al. [46] conducted an empirical analysis of smart contract programming languages based on usability and security from the novice developers' point of view. They considered three programming languages: Solidity, Pact, and Liquidity. The study concluded that although Solidity is the most useful language to a novice developer, it is also the most vulnerable to malicious attacks as novice developers often introduce security vulnerabilities, which can leave the contracts exposed to threats. More recently, in another study, Parizi et al. [47] carried out an assessment of various static smart contract security testing tools for Ethereum and its programming language, Solidity. Their results showed that the SmartCheck tool is statistically more effective than the other automated security testing tools. However, their study considers only the effectiveness, usability, etc. of the tools, but not whether developers are aware of them in practice. Groce et al. [22] summarized the results of security assessments (both manual and automated) performed on smart contracts by a security company. The authors argued that their results pertain to more important contracts (in contrast to prior surveys) since developers were willing to pay for the assessments. Based on the results, they categorized security issues and provided statistics on their frequency, impact, and exploitability. Li et al. [32] studied a wide range of security issues in blockchain technology. They conducted a systematic examination of security risks to blockchain by studying popular blockchain platforms (e.g., Ethereum, Bitcoin, Monero).

5 Discussion and Conclusion

Based on the volume of smart contract related discussions on Stack Exchange (i.e., Stack Overflow and Ethereum Stack Exchange) and Medium, we found that interest in smart contracts—at least from the developers' perspective—seems to have peaked in the first few months of 2018, and has been slowly declining since then. This trend also coincides with a decline in the price of ETH. It will be interesting to see whether this negative trend will continue into the future, or if the decline was just a temporary disillusionment after the initial hype.

We also found that even though most smart contract related questions on Stack Exchange receive at least one answer, extended discussions that would include many answers or comments are rare. The topics of smart contract related discussion on Stack Exchange seem to be dominated by a narrow stack (e.g.,

Solidity, Go Ethereum, Truffle, web3.js), and we observe the prevalence of similar topics on Medium. For example, on both sites, alternative languages (e.g., Vyper) are rarely discussed.

We also observed limited discussion of security-related topics on Stack Exchange, which is very concerning since many smart contracts suffer from security vulnerabilities in practice and since many developers rely on Stack Overflow and similar sites. On Stack Exchange, less than 5% of posts mention security or vulnerabilities; while on Medium, the ratio is around 41%. On Stack Exchange, re-entrancy is the most discussed vulnerability, which seems to be in large part due to the infamous “DAO attack.” Similarly, Stack Exchange posts rarely mention security tools. Further, security tools are even less frequently mentioned in response to question about vulnerabilities (e.g., in conjunction with question about re-entrancy, even though some of the tools can detect re-entrancy vulnerabilities). Fortunately, Medium has a lot more posts that discuss security tools. We find Oyente and Mythril to be the most popular among those tools.

Finally, studying what other topics smart contract developers discuss, we found a significant number of posts about JavaScript and related technologies (and web technologies more generally). This suggests that many smart contract developers have background and interest in JavaScript. We will explore the relationship between the JavaScript and smart-contract developer communities deeper in future work. Similarly, we will also study the relation between smart-contract and safety-critical technologies from the perspective of developers in future work.

Acknowledgment We thank the anonymous reviewers of our manuscript for their insightful comments.

References

1. Stack Exchange Data Explore. <https://data.stackexchange.com/>, accessed on 11/22/2019.
2. di Angelo, M., Salzer, G.: A survey of tools for analyzing Ethereum smart contracts. In: Proc. of the 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON) (April 2019)
3. Ashutosh KS: Top 10 Sites to Ask All Your Programming Questions. <https://www.hongkiat.com/blog/programming-questions-websites/> (2017), accessed on 9/23/2019.
4. Atzei, N., Bartoletti, M., Cimoli, T.: A survey of attacks on Ethereum smart contracts (SoK). In: Proc. of the 6th International Conference on Principles of Security and Trust (POST). pp. 164–186. Springer (April 2017)
5. Bartoletti, M., Pompianu, L.: An empirical analysis of smart contracts: Platforms, applications, and design patterns. In: Proc. of the 1st Workshop on Trusted Smart Contracts (WTSC). pp. 494–509 (April 2017)
6. Barua, A., Thomas, S.W., Hassan, A.E.: What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering* **19**(3), 619–654 (June 2014)

7. Brent, L., Jurisevic, A., Kong, M., Liu, E., Gauthier, F., Gramoli, V., Holz, R., Scholz, B.: Vandal: A scalable security analysis framework for smart contracts. arXiv preprint arXiv:1809.03981 (2018)
8. Chang, J., Gao, B., Xiao, H., Sun, J., Yang, Z.: sCompile: Critical path identification and analysis for smart contracts. arXiv preprint arXiv:1808.00624 (2018)
9. Chen, H., Pendleton, M., Njilla, L., Xu, S.: A survey on Ethereum systems security: Vulnerabilities, attacks and defenses. arXiv preprint arXiv:1908.04507 (2019)
10. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the internet of things. *IEEE Access* **4**, 2292–2303 (2016)
11. Colwell, A.: How Medium became a leading blogging platform – Salesflare blog. <https://blog.salesflare.com/how-medium-became-a-leading-blogging-platform-9b4b08d9d3ac>, accessed on 11/26/2019.
12. ConsenSys: Security tools. Ethereum Smart Contract Best Practices, https://consensys.github.io/smart-contract-best-practices/security_tools/ (2019)
13. Di Angelo, M., Salzer, G.: A survey of tools for analyzing Ethereum smart contracts. In: Proc. of the 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON) (2019)
14. Ducasse, S., Rocha, H., Bragagnolo, S., Denker, M., Francomme, C.: SmartAnvil: Open-source tool suite for smart contract analysis (2019)
15. Ellul, J., Pace, G.: Runtime verification of Ethereum smart contracts. In: Workshop on Blockchain Dependability (WBD), in conjunction with 14th European Dependable Computing Conference (EDCC) (2018)
16. EthereumPrice.org: Ethereum price, charts & history. <https://ethereumprice.org/>, accessed on 12/04/2019.
17. FeedreaderObserve: Ethereum Stack Exchange news. <https://feedreader.com/observe/ethereum.stackexchange.com>, accessed on 12/10/2019.
18. Feist, J., Grieco, G., Groce, A.: Slither: a static analysis framework for smart contracts. In: Proc. of the 2nd IEEE/ACM International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB). pp. 8–15. IEEE (2019)
19. Fenn, J., Raskino, M.: Mastering the hype cycle: how to choose the right innovation at the right time. Harvard Business Press (2008)
20. Finley, K.: A \$50 million hack just showed that the DAO was all too human. *Wired* <https://www.wired.com/2016/06/50-million-hack-just-showed-dao-human/> (June 2016)
21. Grishchenko, I., Maffei, M., Schneidewind, C.: EtherTrust: Sound static analysis of Ethereum bytecode. Tech. rep., Technische Universität Wien (2018)
22. Groce, A., Feist, J., Grieco, G., Colburn, M.: What are the actual flaws in important smart contracts (and how can we find them)? arXiv preprint arXiv:1911.07567 (2019)
23. Harz, D., Knottenbelt, W.: Towards safer smart contracts: A survey of languages and verification methods. arXiv preprint arXiv:1809.09805 (2018)
24. Internet Archive: Stack Exchange Directory Listing. <https://archive.org/download/stackexchange>, accessed on 9/23/2019.
25. IOHK: Plutus: a functional contract platform. <https://testnet.iohkdev.io/en/plutus/>, accessed on 3/11/2020.
26. Jiang, B., Liu, Y., Chan, W.: ContractFuzzer: Fuzzing smart contracts for vulnerability detection. In: Proc. of the 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE). pp. 259–269. ACM (2018)
27. Jiang, H., Liu, D., Ren, Z., Zhang, T.: Blockchain in the eyes of developers (2018)

28. Kalra, S., Goel, S., Dhawan, M., Sharma, S.: ZEUS: Analyzing safety of smart contracts. In: Proc. of the 2018 Network and Distributed Systems Security Symposium (NDSS) (2018)
29. Kirillov, D., Iakushkin, O., Korkhov, V., Petrunin, V.: Evaluation of tools for analyzing smart contracts in distributed ledger technologies. In: Proc. of the 19th International Conference on Computational Science and Its Applications (ICCSA). pp. 522–536. Springer (2019)
30. Krupp, J., Rossow, C.: teEther: Gnawing at Ethereum to automatically exploit smart contracts. In: Proc. of the 27th USENIX Security Symposium. pp. 1317–1333 (2018)
31. Laszka, A., Eisele, S., Dubey, A., Karsai, G., Kvaternik, K.: TRANSAX: A blockchain-based decentralized forward-trading energy exchange for transactive microgrids. In: Proc. of the 24th IEEE International Conference on Parallel and Distributed Systems (ICPADS) (December 2018)
32. Li, X., Jiang, P., Chen, T., Luo, X., Wen, Q.: A survey on the security of blockchain systems. Future Generation Computer Systems (2017)
33. Litan, A., Leow, A.: Hype cycle for blockchain technologies, 2019. Tech. Rep. G00383155, Gartner Research (July 2019), <https://www.gartner.com/en/documents/3947355/hype-cycle-for-blockchain-technologies-2019>
34. Liu, C., Liu, H., Cao, Z., Chen, Z., Chen, B., Roscoe, B.: Reguard: finding reentrancy bugs in smart contracts. In: Proc. of the 40th International Conference on Software Engineering: Companion Proceedings (ICSE). pp. 65–68. ACM (2018)
35. Liu, J., Liu, Z.: A survey on security verification of blockchain smart contracts. IEEE Access **7** (2019)
36. Luu, L., Chu, D.H., Olickel, H., Saxena, P., Hobor, A.: Making smart contracts smarter. In: Proc. of the 23rd ACM SIGSAC Conference on Computer and Communications Security (CCS). pp. 254–269 (October 2016)
37. Mavridou, A., Laszka, A.: Designing secure Ethereum smart contracts: A finite state machine based approach. In: Proc. of the 22nd International Conference on Financial Cryptography and Data Security (FC) (February 2018)
38. Mavridou, A., Laszka, A.: Tool demonstration: FSolidM for designing secure Ethereum smart contracts. In: Proc. of the 7th International Conference on Principles of Security and Trust (POST) (April 2018)
39. Mavridou, A., Laszka, A., Stachtari, E., Dubey, A.: VeriSolid: Correct-by-design smart contracts for Ethereum. In: Proc. of the 23rd International Conference on Financial Cryptography and Data Security (FC) (February 2019)
40. Mossberg, M., Manzano, F., Hennenfent, E., Groce, A., Grieco, G., Feist, J., Brunson, T., Dinaburg, A.: Manticore: A user-friendly symbolic execution framework for binaries and smart contracts. arXiv preprint arXiv:1907.03890 (2019)
41. Mueller, B.: Smashing Ethereum smart contracts for fun and real profit. 9th Annual HITB Security Conference (HITBSecConf) (2018)
42. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
43. Newman, L.H.: Security news this week: \$280m worth of Ethereum is trapped thanks to a dumb bug. WIRED, <https://www.wired.com/story/280m-worth-of-ethereum-is-trapped-for-a-pretty-dumb-reason/> (November 2017)
44. Nikolić, I., Kolluri, A., Sergey, I., Saxena, P., Hobor, A.: Finding the greedy, prodigal, and suicidal contracts at scale. In: Proc. of the 34th Annual Computer Security Applications Conference (ACSAC). pp. 653–663 (2018)
45. O’Connor, R.: Simplicity: A new language for blockchains. In: Proc. of the 2017 Workshop on Programming Languages and Analysis for Security. pp. 107–120. PLAS (2017)

46. Parizi, R.M., Amritraj, Dehghantanha, A.: Smart contract programming languages on blockchains: An empirical evaluation of usability and security. In: Proc. of the 1st International Conference on Blockchain (ICBC). pp. 75–91 (2018)
47. Parizi, R.M., Dehghantanha, A., Choo, K.K.R., Singh, A.: Empirical vulnerability analysis of automated smart contracts security testing on blockchains. In: Proc. of the 28th Annual International Conference on Computer Science and Software Engineering (CASCON) (2018)
48. Permenev, A., Dimitrov, D., Tsankov, P., Drachsler-Cohen, D., Vechev, M.: VerX: Safety verification of smart contracts. In: Proc. of the 41st IEEE Symposium on Security and Privacy (S&P) (2020)
49. Solidity Documentation: Security considerations – use the Checks-Effects-Interactions pattern. <http://solidity.readthedocs.io/en/develop/security-considerations.html#use-the-checks-effects-interactions-pattern> (2018), accessed on 9/23/2019.
50. Stortz, R.: Rattle – an Ethereum EVM binary analysis framework. REcon Montreal (2018)
51. Tapscott, A., Tapscott, D.: How blockchain is changing finance. *Harvard Business Review* **1**(9) (2017)
52. Tikhomirov, S., Voskresenskaya, E., Ivanitskiy, I., Takhaviev, R., Marchenko, E., Alexandrov, Y.: Smartcheck: Static analysis of ethereum smart contracts. In: Proc. of the 1st IEEE/ACM International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB). pp. 9–16. IEEE (2018)
53. Torres, C.F., Schütte, J., et al.: Osiris: Hunting for integer bugs in Ethereum smart contracts. In: Proc. of the 34th Annual Computer Security Applications Conference (ACSAC). pp. 664–676. ACM (2018)
54. Tryfanava, D.: 25 Best Active Forums for Programmers. <https://vironit.com/best-active-forums-for-programmers/> (2018), accessed on 9/23/2019.
55. Tsankov, P., Dan, A., Cohen, D.D., Gervais, A., Buenzli, F., Vechev, M.: Securify: Practical security analysis of smart contracts. In: 25th ACM Conference on Computer and Communications Security (CCS) (2018)
56. Underwood, S.: Blockchain beyond Bitcoin. *Communications of the ACM* **59**(11), 15–17 (2016)
57. Vyper Team: Vyper documentation. <https://vyper.readthedocs.io/en/latest/>, accessed on 3/11/2020.
58. Wang, H., Li, Y., Lin, S.W., Ma, L., Liu, Y.: VULTRON: catching vulnerable smart contracts once and for all. In: Proc. of the 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER). pp. 1–4 (2019)
59. Wang, N., Zhou, X., Lu, X., Guan, Z., Wu, L., Du, X., Guizani, M.: When energy trading meets blockchain in electrical power system: The state of the art. *Applied Sciences* **9**(8) (2019). <https://doi.org/10.3390/app9081561>
60. Wöhrer, M., Zdun, U.: Design patterns for smart contracts in the ethereum ecosystem. In: Proc. of the 2018 IEEE Conference on Blockchain. pp. 1513–1520 (2018)
61. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Tech. Rep. EIP-150, Ethereum Project – Yellow Paper (April 2014)
62. Zhou, E., Hua, S., Pi, B., Sun, J., Nomura, Y., Yamashita, K., Kurihara, H.: Security assurance for smart contract. In: Proc. of the 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS). pp. 1–5. IEEE (2018)
63. Zhu, L., Zheng, B., Shen, M., Yu, S., Gao, F., Li, H., Shi, K., Gai, K.: Research on the security of blockchain data: A survey. arXiv preprint arXiv:1812.02009 (2018)