A Novel Reputation System for Mobile App Stores Using Blockchain

Gonçalo Sousa Mendes and Daniel Chen, Aptoide S.A.
Bruno M.C. Silva, Universidade da Beira Interior
Carlos Serrão, Information Sciences and Technologies and Architecture Research Center
João Casal, Aptoide S.A.

With thousands of mobile applications submitted to online application stores, the mobile application market has experienced significant growth. This growth is accompanied by an increase in malware presence. A possible solution would be to leverage those reports across all mobile ecosystems, creating a shared reputation system.

The mobile applications economy is an expanding market. In August 2019, the Google Play Store had almost 2.5 million apps in its repositories, the Apple App Store had 1.8 million, and Aptoide had nearly 1 million. There were 21.3 billion downloads in Google Play alone, with several other third-party app stores experiencing similar rates. However, malware presence has also increased, which exposes users to possible privacy breaches and loss of assets, for example, baking Trojans to steal login credentials, as app stores are struggling to keep up with such malware growth.

Current centralized models of cybersecurity verification and validation hardly fit the ever-growing global mobile app economy size. Several apps have obfuscated...
code that hardens automated code analysis. Studying their behavior during execution is not feasible due to the prohibitive costs and high inefficiency. This shows how a centralized approach to cybersecurity may not be the only answer to fight mobile malware, where each app store has its own methods and does not share the discovered threats.

A new approach to this problem is to relate the multiple actions of millions of users and thousands of developers (reputation events), such as ratings, comments, or app uploads, to a degree of trust and use that found reputation in the cybersecurity processes to provide trustworthy and relevant decision support to the quality assurance (QA) and security teams. Moreover, such an approach could benefit both the developers as they could switch between different markets while keeping their reputation and the consumers as it makes it easier for them to trust a new developer. This would significantly increase the content quality of all app stores, having a direct impact in their revenue, a practice that is widely adopted on e-commerce platforms, such as Amazon or eBay, among several others.4,5

Building a consortium solution for several competing app stores presents several problems of performance, scalability, interoperability, data security, and trust between the entities.6–9 These problems include 1) trust between the app stores inside of the consortium; 2) the scalability, interoperability, and performance to manage multiple actions of millions of users and thousands of developers, where several different app stores can contribute, and each one has its own business model and rules; 3) data privacy as the user’s data need to be securely saved to comply with data regulations; 4) traceability as it is necessary to know the history of actions of the users; and 5) cost-efficiency.

Blockchain is a distributed ledger in a peer-to-peer network with unique characteristics that can effectively solve the security problems.6 The new blocks are validated through a consensus protocol that also guarantees a total order of them. Blockchains can be public, private, or a mixture of both. In public blockchains, everyone can join the network and participate. Every node can mine the new block and is rewarded by the issuers of transactions inside that block. In private blockchains, only one entity controls the network and dictates who can join the network and read and write blocks. In this network, it is not necessary to pay mining fees. In consortium blockchains, the control of the network is shared among all participant entities, where everyone can read and write, but joining requires the approval of all members. As in private blockchains, there are no mining fees.

A possible solution to tackle the mentioned problems consists of using a blockchain integrated with a cloud system to support the blockchain with the necessary logic operations and provide an interface for the QA and security teams.

Aptoide, already considered one of the safest Android app stores,2 has nonetheless felt the need to improve its security by developing such a system. Here, millions of users, through the app stores installed on their smartphones, perform reputation events that are disseminated through the blockchain’s services. The app stores can then infer the threats on these applications through those reputation events and remove them, thus creating a mobile cyberphysical system (CPS) connecting millions of devices with blockchain and cloud/interface services.

This article presents a cloud-based solution to manage reputation events that are integrated with a consortium blockchain, to serve as a distributed ledger to store those events as the support for the reputation system. This combination allows for the identification of compromised apps through a dedicated interface for the analysts, where the apps classified as a threat by the high reputation users are displayed first for review. This provides an improvement on QA to users and mitigates the potential for privacy breaches and data theft for all users of the app stores who wish to be in the consortium. This possibility generates a global app economy with increased security, trust, and
transparency, where each app store can control how the reputations are calculated, allowing for a better fit on its business model.

The main contributions of this work are

1. the study and implementation of a cloud solution integrated with a blockchain to manage and store millions of reputations events
2. the implementation of a real case study to evaluate the performance of the solution developed.

From the obtained evaluation results, it was observed how the combination of a cloud solution with a blockchain could improve this type of system regarding data security, interoperability, and scalability, while being cost-effective.

RELATED WORK AND BACKGROUND

Some works have already integrated blockchain in their reputation systems to overcome the problems mentioned.\textsuperscript{8,10} Some examples follow.

Rep on the Roll\textsuperscript{11} is a generalist reputation system for peer-to-peer networks, where blockchain is used for effective communication and information sharing between clients and whenever behavior traceability is desired. Both the trust and reputation of peers are maintained and unmodified due to the proprieties of blockchain. The authors compared their work with eBay, which can process an average of 23,184 transactions per second (TPS), against the modest quantity of 10 in their work. Such a low number comes from the time it takes to mine a block in a bitcoin-based blockchain. This is a performance problem as well as an opportunity for a denial of service attack, as the authors noted. This performance issue also makes the network hard to adopt and implement on larger scales since the resources required from each node are costly. The network is also public, which raises some data privacy concerns as all of the user’s data are accessible on a public network.

SH-BlockCC\textsuperscript{8} proposes a cloud solution based on blockchain to increase the security and trust of the Internet of Things (IoT) in smart homes while maintaining availability, scalability, and traceability. The authors demonstrated how their proposed architecture could make the IoT more secure and efficient. However, their blockchain is too tailored to their problem without providing clear implementation details, making it very difficult to translate it to other domains.

Ink Protocol\textsuperscript{12} is a decentralized third-party software to handle a payment system for e-commerce marketplaces. The goal is to increase trust—and therefore, security—by evaluating the transactions between the sellers and buyers to create reputation scores. This is very close to our goal. However, they require cryptocurrency, so the software is based on Ethereum, a public blockchain. As mentioned, this type of blockchain brings some performance and data privacy issues as well as the necessary mining fees.

Decentralized Science\textsuperscript{13} is a blockchain-based distributed platform for science publications, together with a reputation system for peer reviewers. The platform is used to store the peer review process communications, where the smart contracts validate the system’s rules. This system is also based on Ethereum.

Some other implementations and areas use the blockchain for storage and reputation management, such as an emissions trading scheme\textsuperscript{14} with a private blockchain, where new participants must pay to adhere; the automobile industry, where a public blockchain is used with an external service to provide the necessary security and data privacy; and CPS\textsuperscript{16} a bitcoin-based blockchain with extra work to offer the required CPS systems data privacy, among others.

None of the presented works can provide a truly scalable solution while ensuring data privacy. A possible solution is to use a consortium blockchain, which is a blockchain that is controlled by a set of trusted entities, not entirely public or entirely private. This approach brings several benefits to the collaborations between organizations besides security, namely, scalability (it is easy to add new nodes and other app stores) and data privacy (only authorized app stores can access the data).\textsuperscript{7,17} The solution is still cost-effective\textsuperscript{7,8,18} since there is no need to pay the mining fees as is done for public blockchains, and the maintenance costs can be equally shared between all participant entities. The consortium approach also adds the necessary trust between entities in the app market since no one fully controls the network or can tamper with the data unnoticed.

Some works already demonstrate the usefulness of such an approach,\textsuperscript{4} where the authors used a consortium blockchain combined with a common database for asset trading. Here, the reputation is computed inside of the network on a smart contract. This makes it impossible to adjust to the specific needs of each new entity, a desirable property for the app market ecosystem. Also, the use of a common database demonstrates the need for a balance between the cloud and a blockchain.
To implement a consortium blockchain, we selected the Hyperledger Fabric (HLF), a blockchain framework under the umbrella of the Hyperledger Greenhouse, hosted by The Linux Foundation. It was chosen due to its strong modularity, widespread adoption, popularity, and proven scalability properties, besides providing all of the mentioned requirements.

HLF has three distinct phases. The execute phase, which starts when a client submits a new transaction to the endorsement peers, runs the chaincode (equivalent to smart contracts), creating the read–write (RW) sets. Defined by the endorsement policy, these peers are the only ones that can simulate new transactions. This set is then returned to the client, which aggregates the necessary RW sets and sends them to the ordering service, starting the order phase. Here, through a consensus protocol, a new block is created, defining the total order of transactions, and sent to all peers. In the validate phase, the peers verify the latest transactions and update their world state. This world state is a snapshot of the most recent state of the blockchain.

To improve the data security and compliance with data regulations, Fabric provides data privacy. In this mechanism, data are written in a separated database, are accessible only to the authorized peers, and only the hash of the transaction is saved on the blocks. This allows for personal and sensitive data to be deleted as only the hash is permanently kept. HLF also contains the Membership Service Provider, a service responsible for defining the roles of the entities inside of the network and validating the certificates generated by the Fabric Certificate Authority that is abstracted through this service.

It is through those services that Fabric provides the necessary data privacy and security to the blockchain solution.

**A SCALABLE BLOCKCHAIN-BASED SOLUTION FOR REPUTATION MANAGEMENT**

The proposed system architecture is composed of two major components: the blockchain, where all of the reputation events are stored, and the cloud infrastructure, which is responsible for reputation management and the app cybersecurity processes. The cloud infrastructure supports an interface that provides the QA and security teams with the necessary dashboards for their analysis, creating an intelligent decision support system.

This cloud is only a complementary module, provided as software as a service, since the blockchain is enough to store the reputation events. However, through the cloud it is possible to deliver a dedicated interface to the QA and security teams to analyze the apps considered dangerous by the reports provided by the users, thus improving the cybersecurity processes of app stores. The cloud is app store agnostic, a reputation module, provided as software as a service, since the blockchain is enough to store the reputation events. However, through the cloud it is possible to deliver a dedicated interface to the QA and security teams to analyze the apps considered dangerous by the reports provided by the users, thus improving the cybersecurity processes of app stores. The cloud is app store agnostic, a reputation module, provided as software as a service, since the blockchain is enough to store the reputation events. However, through the cloud it is possible to deliver a dedicated interface to the QA and security teams to analyze the apps considered dangerous by the reports provided by the users, thus improving the cybersecurity processes of app stores.

To guarantee that the blockchain is app store agnostic, a reputation event that is as generic as possible was defined, reducing the complexity of the smart contract (that is, the chaincode in our blockchain implementation), making it easier for any app store to model them following their business model. By proving an external application programming interface (API), this interoperability is improved as it dismisses the complicated knowledge to communicate directly with the blockchain. This API has four operations: 1) insert event: add new reputation events; 2) get event: get a specific event; 3) get by date: fetch all events that were added to the blockchain on a particular day; and 4) get by user: returns all events from a user.

To guarantee that the blockchain is app store agnostic, a reputation event that is as generic as possible was defined, reducing the complexity of the smart contract (that is, the chaincode in our blockchain implementation), making it easier for any app store to model them following their business model. By proving an external application programming interface (API), this interoperability is improved as it dismisses the complicated knowledge to communicate directly with the blockchain. This API has four operations: 1) insert event: add new reputation events; 2) get event: get a specific event; 3) get by date: fetch all events that were added to the blockchain on a particular day; and 4) get by user: returns all events from a user.
have a unique ID to place at the beginning of the ID field.

The default HLF Certificate Authority was used to generate X509 certificates. As for the consensus algorithm, we used Raft. This protocol works in a leader-follower fashion, is crash fault tolerant, and can be distributed in different servers and organizations, thus decentralizing the control of the consensus algorithm. Raft is the first fully consensus protocol integrated in Fabric (implemented in V1.4.1, April 2019), therefore providing fast integration and a higher performance when compared to its ancestor, a junction of Kafka and ZooKeeper.

All Fabric components run in containers and communicate through Remote Procedure Calls over a secure and authenticated channel using Transport Layer Security protocol. The chain-code was written in Go, and our endorsement policy stated that at least one peer of each organization (randomly chosen by Fabric) has to endorse the transaction.

The version of the HLF used is 1.4.1. For the implementation and our proof of concept, two organizations were used for deployment, the real app store, Aptoide, and a dummy app store, named app store X, to simulate the dissemination of information in the blockchain.

Cloud implementation

The proposed cloud solution, integrated with the blockchain, serves the purpose of supporting the described intelligent decision support system. It is microservice oriented and built on top of two main modules: Event Tracker and Data Collection (Figure 1).

The Event Tracker module is responsible for fetching the events from the app store, analyzing each one individually, and rerouting them to the blockchain (Figure 1). If the event is a flag (Aptoide allows users to flag its app), it is stored in a local

![FIGURE 1. A flow chart of the implemented cybersecurity process, which depicts the flow of the two main off-chain modules and their interactions with the blockchain. It also demonstrates how the reputations are used to update the app threat levels.](image)
FIGURE 2. The tests of the HLF configuration, with the throughput and latency for each analysis performed. (a) and (b) Results while varying the block size; 2 MB was chosen as a safe threshold. (c) and (d) Results while changing the number of messages per block; 100 messages were selected. (Continued)
immediate action, the flags given by high-reputation users and uploads by low-rated developers are also treated separately to update the app threat level and the list of apps submitted by low-rated developers, respectively.

This app threat level represents the severity deemed by the system for each app, increasing or decreasing its urgency in being reviewed, making the system react faster to what may be the biggest threats to the end users.

The Data Collection module (Figure 1) is responsible for collecting the events from the blockchain, including from other stores as well. During this phase, the reputation events are processed through an internally defined

---

**FIGURE 2. (Continued)** (e) and (f) Results on the variation on the number of peers. The initial configuration has the best performance. (g) and (h) Results on the variation on the number of raft nodes. No changes in the performance were detected.
reputation system, and the reputation of the users involved is updated. If the user is new on Aptoide, we check to see if he has already done other actions on other app stores, updating his reputation accordingly. Then, the system updates the threat level for each app with the updated reputations.

Each module is executed once per day by a scheduler, given that the Event Tracker always starts first at the beginning of the day (after midnight in UTC) by fetching the events from the previous day and sending them to the blockchain through the add_event functionality. However, if it is deemed necessary (for example, by an increase in flag actions), it can be executed more frequently, further decreasing the reaction time. The Data Collection module then fetches all of the events from all of the app stores from the previous day using the get_event_by_date functionality. This query is done on the date from which the events were added to the blockchain to avoid missing events added by other app stores that occurred before the previous day but were all added on that last day.

EVALUATION RESULTS
The evaluation consisted of several tests. First, we tested the HLF configuration to benchmark and test the performance, scalability, and fault tolerance. Then, we tested the off-chain solution to evaluate the capacity of the system in managing several thousands of events. All of the tests were performed on a Dell R210-2 remote server with an Intel Quad-Core Xeon E3-1270v2, with 16-GB DDR3 of RAM and 2-TB SATA2 for the hard drive, running Ubuntu 16.04 LTS.

HLF evaluation
Three different steps in the evaluation were defined. First, the performance was evaluated by measuring the impact of changing first the block size and then the number of transactions on each block. Second, the scalability was evaluated by adding more peers to the organizations, and third, the fault tolerance was evaluated. For each test, two metrics are presented, the latency (time of each operation to complete) and throughput (TPS). Moreover, the tests do not include the authentication of users, which is one of the most cumbersome steps as it is processed outside of HLF itself. The experiments were performed by Hyperledger Caliper, a benchmark tool for Hyperledger blockchains.

In this scenario, three functionalities were tested: a write operation (add_event), a read operation (get_by_id), and a read operation that returns several events (get_by_date). For each event, 1,000 different transactions were sent at a rate of 100/s. The fields in the events were generated randomly (aside from the ID) and incrementally, starting at one, and the date was chosen randomly for a set of 10 different dates. The maximum number of transactions to be sent per second was 100. The initial configuration contained two peers in each organization and three raft nodes. Figure 2 contains the results of the tests, where each plot contains the results for all three operations.

Aside from the block size, we used the initial configuration provided by the HLF. Figure 2(a) and (b) indicates that both the latency and TPS are somewhat stable through the trials, with a small peak in TPS for the add_event operation at 2 MB. This cap is explained by the limitation on the number of transactions. Therefore, we increasingly changed this limitation to observe its behavior. From Figure 2(c) and (d), it is observable how incrementing the number of transactions improves the performance. Latency wise, further increases are not observable after 75 transactions per block. As for the TPS, it reaches its peak at 100 transactions per block, with approximately 70 TPS for write operations. It is here that the network reaches its saturation and is not capable of processing more transactions.

The scalability of the system was tested by iteratively adding peers to each organization. The initial configuration is the one with the best results [Figure 2(e) and (f)]. These results are not expected as having more peers means more availability. However, an observation on the resource consumption explains this performance, as the tests are run in a single server that rapidly reaches its limit, and the Caliper must wait for all of the peers to commit the new block.

Finally, the system was tested regarding its fault tolerance [Figure 2(g) and (h)]. From the results, it is possible to conclude that adding fault tolerance does not create performance issues. This is expected, as the messages are only sent to the leader, which simultaneously propagates the message and waits for a majority of the responses. Therefore, as long the number of nodes does not consume all of the server resources, it is not expected that we would observe changes in the times reported. For the following tests, the number of raft nodes was set to three.

The configuration found is useful for our setup but may not be the best for others. However, these tests show how versatile and scalable the HLF is, where it is possible to add or remove peers in any location due to its modularity.

Cloud evaluation
After finding the best configuration for HLF, the whole system was...
tested by generating 10,000 reputation events (a number way above the average daily in Aptoide) for 1,000 users (that were previously inserted in the database), corresponding to a fictional day. Using the flow in Figure 1, the results presented here include pre-processing the data, sending events to the blockchain, requesting those events, and updating the reputations and app threat levels. Note that, despite all of the components being in the same network, the requests are not made in local-host to approximate the results with a real-world scenario without adding too much latency.

Preprocessing the data and sending them to the blockchain is continuous; however, run times are presented separately to facilitate the analysis of each module. The full process for the 10,000 events took approximately 11 h. The initial flow of the cloud takes the most time, having 1 h in preprocessing and almost 10 h in sending them to the blockchain. The processes of preparing and analyzing the data are extremely fast, demonstrating the ability to quickly react to new threats while preparing the events.

Sending the events to the blockchain is where the system spends most of its time, averaging 3 s per event, which includes the latency of the requests, authentication (done for every new request), HLF internal flow, and the response, all of which become the bottleneck of our system. Given the added layer provided by the blockchain, with the extra steps of security and replication and the fact that our implementation is a proof of concept that does not consider parallelization strategies, these times are not a surprise and confirm that the system can be improved to leverage the capabilities of HLF and deliver higher performances.

To request the 10,000 events from the blockchain, it only took 3 s, which is a fast response considering the number of events and the times to send them. Then, it took almost 18 min to process all of those events and update the reputations and app threat levels. This demonstrates how fast the off-chain solution can collect the events and process them, reacting rapidly to

ABOUT THE AUTHORS

**GONÇALO SOUSA MENDES** is a researcher at Aptoide S.A., Lisbon, 1600-196, Portugal. His research interests include artificial intelligence, cloud development, and high-performance computing. Mendes received his M.Sc. in machine learning from Faculdade de Ciências e Tecnologias da Universidade Nova de Lisboa. Contact him at goncalo.mendes@aptoide.com.

**DANIEL CHEN** is a researcher at Aptoide S.A., Lisbon, 1600-196, Portugal. His research interests include cloud development and mobile applications. Chen received his B.Sc. in computer science from ISCTE–Instituto Universitário de Lisboa. Contact him at daniel.chen@aptoide.com.

**BRUNO M.C. SILVA** is an assistant professor and the head of technology at the Faculty of Design, Technology and Communication of the European University, 6200, Covilhã, Portugal. His research interests include delay-tolerant networks, vehicular networks, mobile computing, e-health, the Internet of Things, and ambient assisted living. Silva received his Ph.D. in computer science from the Universidade da Beira Interior. Contact him at bruno.silva@it.ubi.pt.

**CARLOS SERRÃO** is an assistant professor at the Information Sciences and Technologies Department, ISCTE–Instituto Universitário de Lisboa, Lisboa, 1649-026, Portugal. His research interests include distributed applications and systems, information security, and mobile and web information security. Serrão received his Ph.D. in computer architecture and distributed systems from Universitat Politècnica de Catalunya. He is a member of the Association for Computing Machinery, Open Web Application Security Project and Associação Portuguesa para a Promoção da Segurança de Informação. Contact him at carlos.serrao@iscte-iul.pt.

**JOÃO CASAL** is the head of research and development at Aptoide S.A., Lisbon, 1600-196, Portugal. His research interests encompass several tech trends that relate to mobile and ubiquitous computing, from cybersecurity to artificial intelligence, the Internet of Things, and blockchain. Casal received an M.Sc. in systems engineering from the Universidade do Minho. Contact him at joaocasal@gmail.com.
the changes in the users’ reputations and reflecting those same changes in the cybersecurity processes as the app’s threat level.

This article presents a blockchain-based solution to manage reputation events through an HLF named TrustChain. This system provides the necessary data privacy and security but also guarantees scalability without a significant loss of performance while also being cost-effective. This work includes an experimental analysis and evaluation of the proposed system using a real app store, Aptoide, together with an explanation of how the solution was constructed and implemented to achieve fast response times regarding mobile cybersecurity.

It was demonstrated how a modular blockchain, through Fabric, is adaptable and can easily scale. Both results validate the solution, attesting that it does execute as intended in reacting to new threats and sharing, in a secure manner, the events that lead to those reactions. However, it does need some parallelization strategies to better achieve its full potential. Future work includes scaling, with more parallelization strategies and spreading the peers through different servers. Additionally, future work should improve the solution introduced here, either using parallelization strategies or the spreading of peers through different nodes, and use the performance demonstrated as a baseline for benchmarking.

ACKNOWLEDGEMENTS
This work is part of the TrustChain project, cofunded by Programa Operacional Regional de Lisboa [Portugal 2020/European Union (EU)], in the context of the Portuguese Sistema de Incentivos à I&DT Empresarial (project LISBOA-01—0247-FEDER-038315). This work is funded by the Foundation for Science and Technology and the Portuguese Ministry of Science, Technology, and Higher Education through national funds and, when applicable, cofunded EU funds under the project UIDB/EEA/50008/2020.

REFERENCES
1. J. Clement. “Mobile app usage: statistics & facts.” https://www.statista.com/topics/1002/mobile-app-usage/ (accessed Dec. 23, 2019).
2. Y. Ishii et al., “Understanding the security management of global third-party android market places,” in Proc. 2nd ACM SIG-SOFT Int. Workshop App Market Anal.—WAMA 2017, pp. 12–18. doi: 10.1145/3121264.3121267.
3. “Mcafee mobile threat report q1,” Tech. Rep. McAfee, San Jose, CA, 2019. [Online]. Available: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-mobile-threat-report-2019.pdf
4. H. Huang, J. Cai, and S. Xie, “Implementing an asset trading system based on blockchain and game theory,” in Proc. 2019 Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov. (CyberC), pp. 219–226. doi: 10.1109/CyberC.2019.00045.
5. M. Möhlmann, T. Teubner, and A. Graul, “Leveraging trust on sharing economy platforms: Reputation systems, blockchain technology and cryptocurrencies,” in *Handbook of the Sharing Economy*. Cheltenham, U.K.: Edward Elgar, 2019, pp. 290–302.
6. J. Chen, Z. Lu, and H. Song, “Design of personnel big data management system based on blockchain,” Future Gener. Comput. Syst., vol. 101, pp. 1122–1129, Dec. 2019. doi: 10.1016/j.future.2019.07.037.
7. E. Androulaki, C. Cachin, A. D. Caro, and E. Kokoris-Kogias, “Channels: Horizontal scaling and confidentiality on permissioned blockchains,” in *Computer Security*. Berlin: Springer-Verlag, 2018, pp. 111–131.
8. S. Singh, I.-H. Ra, W. Meng, M. Kaur, and G. H. Cho, “SH-BlockCC: A secure and efficient internet of things smart home architecture based on cloud computing and blockchain technology,” *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 4, p. 155014771984415, 2019. doi: 10.1177/1550147719844159.
9. E. Karafiloski and A. Mishev, “Blockchain solutions for big data challenges: A literature review,” in *Proc. IEEE EUROCON 2017–17th Int. Conf. Smart Technol.*, pp. 763–768. doi: 10.1109/EUROCON.2017.8011213.
10. H. Wang, Q. Wang, and D. He, “Blockchain-based private provable data possession,” *IEEE Trans. Dependable Secure Comput.*, early access, Oct. 28, 2019. doi: 10.1109/TDSC.2019.2949809.
11. R. Dennis and G. Owenson, “Rep on the roll: A peer to peer reputation system based on a rolling blockchain,” *Int. J. Digit. Soc.*, vol. 7, no. 1, pp. 1123–1134, Mar. 2016. doi: 10.20533/ijds.2040.2570.2016.0137.
12. Ink, “Decentralized reputation and payments for peer-to-peer marketplaces,” Tech. Rep., Santa Clara, CA, 2018. [Online]. Available: https://paywithink.com/wp-content/uploads/2018/02/Ink_Protocol_Whitepaper_V8_Listia_Inc.pdf
13. A. Tenorio-Fornés, V. Jacynycz, D. Llop-Vila, A. Sánchez-Ruiz, and S. Hassan, “Towards a decentralized process for scientific publication and peer review using blockchain and IPFS,” in *Proc. 52nd Hawaii Int. Conf. Syst. Sci.*, 2019. doi: 10.24251/HICSS.2019.560. [Online]. Available:
14. K. N. Khaki, J. J. Sikorski, K. Hadjinoato, and M. Kraft, “Incorporating seller/buyer reputation-based system in blockchain-enabled emission trading application,” Appl. Energy, vol. 209, pp. 8–19, Jan. 2018. doi: 10.1016/j.apenergy.2017.10.070.

15. Z. Lu, Q. Wang, G. Qu, and Z. Liu, “BARS: A blockchain-based anonymous reputation system for trust management in VANETs,” in Proc. 17th IEEE TrustCom/12th IEEE BigDataSE, Aug. 2018, pp. 98–103. doi: 10.1109/TrustCom/BigDataSE.2018.00025.

16. Y. Zhao, Y. Li, Q. Mu, B. Yang, and Y. Yu, “Secure pub-sub: Blockchain-based fair payment with reputation for reliable cyber physical systems,” IEEE Access, vol. 6, pp. 12,295–12,303, Jan. 2018. doi: 10.1109/ACCESS.2018.2799205.

17. O. Dib, K.-L. Brousiche, A. Durand, E. Thea, and E. Hamida, “Consortium blockchains: Overview, applications and challenges,” Int. J. Advances Telecommun., Sept. 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02271063

18. T. T. Anh Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, “Untangling blockchain: A data processing view of blockchain systems,” IEEE Trans. Knowl. Data Eng., vol. 30, no. 7, pp. 1366–1385, July 2018. doi: 10.1109/TKDE.2017.2781227.

19. E. Androulaki et al., “Hyperledger fabric: A distributed operating system for permissioned blockchains,” in Proc. 13th EuroSys Conf., New York, ACM, 2018, pp. 30:1–30:15. doi: 10.1145/3190508.3190538.

20. A. Baliga, N. Solanki, S. Verekar, A. Pednekar, P. Kamat, and S. Chatterjee, “Performance characterization of hyperledger fabric,” in Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT), June 2018, pp. 65–74. doi: 10.1109/CVCBT.2018.00013.

IEEE Pervasive Computing seeks accessible, useful papers on the latest peer-reviewed developments in pervasive, mobile, and ubiquitous computing. Topics include hardware technology, software infrastructure, real-world sensing and interaction, human-computer interaction, and systems considerations, including deployment, scalability, security, and privacy.