LEDGER
ledgerjournal.org

ISSN 2379-5980 (online)
associated article DOI 10.5195/LEDGER.2022.230

HLF-Kubed: Blockchain-Based Resource Monitoring for Edge Clusters

Achilleas Tzenetopoulos,* Dimosthenis Masouros, Nikolaos Kapsoulis,† Antonios Litke,‡ Dimitrios Soudris, Theodora Varvarigou§

Reviewers: Reviewer A, Reviewer B

Abstract. The final version of the paper "HLF-Kubed: Blockchain-Based Resource Monitoring for Edge Clusters" can be found in Ledger Vol. 7 (2022) 1-16, DOI 10.5195/LEDGER.2022.230. There were two reviewers involved in the review process, neither of whom has requested to waive their anonymity at present, and are thus listed as Reviewers A and B. After initial review by Reviewers A and B, the submission was returned to the authors with feedback for revision (1A). The authors responded (1B) and resubmitted their work. After subsequent evaluation by Reviewer A (2A), revisions made were deemed sufficient to address any concerns, thus ending the peer review process. Author responses have been bulleted for reader clarity.

1A. Review

Reviewer A

Does this paper represent a novel contribution to cryptocurrency or blockchain scholarship?

Not sure

If you answered "yes" to the previous question, in one sentence, describe in your own words the novel contribution made by this paper:

I answered "not sure" because the paper discusses an application of blockchain technology, and shows how a DLT product can run on constrained devices. However, I don't see a novel

* 0x6730CCF240EBe438b3a45C81A3a7640F4d7896C8
† A. Tzenetopoulos(atzenetopoulos@microlab.ntua.gr), D. Masouros (dmasouros@microlab.ntua.gr), and N. Kapsoulis (nkapsoulis@mail.ntua.gr) are PhD candidates in the Department of Electrical and Computer Engineering, NTUA, Athens, Greece.
‡ A. Litke, PhD (litke@mail.ntua.gr) is a Senior Researcher in the Department of Electrical and Computer Engineering, NTUA.
§ D. Soudris (dsoudris@microlab.ntua.gr) and T. Varvarigou (dora@telecom.ntua.gr) are Professors in the Department of Electrical and Computer Engineering, NTUA.

contribution on blockchain concepts because the authors do not explain why, what and how they changed the HLF implementation.

Is the research framed within its scholarly context and does the paper cite appropriate prior works?

Yes

Please assess the article's level of academic rigor.

Good (not excellent but a long way from poor)

Please assess the article's quality of presentation.

Excellent (the motivation for the work is clear, the prose is fluid and correct grammar is used, the main ideas are communicated concisely, and highly-technical details are relegated to appendixes).

How does the quality of this paper compare to other papers in this field?

Top 50%

Please provide your free-form review for the author in this section.

The paper is written very nicely, and it is structured in a good manner.

The reviewer believes that the largest potential for improvement is the motivation and clear justification of the “why blockchain?” and “how is blockchain beneficial” questions:

• Resource monitoring is not the type of data where tamper-proofness, trustworthiness and other DLT core properties are of highest importance. What is the advantage of DLT-based monitoring compared to, for example, MoM-based data collection? With n nodes, the amount of raw monitoring data is stored n-fold; maybe it is more economical to reduced from n to a smaller number, e.g. “max of 4”?

• Why is consensus over raw monitoring data necessary at all? The consensus-participating edge nodes/organizations cannot verify the accuracy of the raw data because that data is only verifiable on the edge device itself.

• Consensus-finding incurs overheads (especially during the coordination), how does that overhead relate to the claim that a DLT-based solution is avoiding the downsides of the centralized data collection? (in the paper’s abstract, the authors say that the centralized data collection approaches “are unable to handle the increased load”)

• The blockchain network is technically and organizationally decentralized, but the network is logically centralized as it has a shared single truth, completely replicated across nodes (with some delays, and network outages). How do edge devices (which are not only “HLF clients” but actual HLF nodes) choose which other HLF node(s) to gossip to?

ISSN 2379-5980 (online)
associated article DOI
10.5195/LEDGER.2022.230

• How does a DLT-based solution compare (conceptually) to a load-balanced cluster of monitoring servers, with the monitoring subjects choosing a server randomly on each report?

• With unbounded ledger growth and no deletion facilities in HLF, how is the stored data rotated (i.e. discarded)?

Another important aspect which needs improvement is the case study with measurements. The reviewer believes that to go beyond 3 edge devices (which is way too few), a simulated (e.g. cloud-based) IoT network is useful (but also easy to set up, because the virtual devices can be bootstrapped based on the same template as for RasPi devices). Following the paper's 5-sec measurement interval, one should be able to see the effects of both spikes (many edge devices sending out data) and flats (average load). In particular, a larger number of edge devices would enable conclusions about throughput and latency, and not just about processing time and overheads.

Figure 4 needs a rework to clearly display which functionality is deployed on edge device and which is not.

Algorithm 1 (why not provide Golang chaincode source code, too?) appears not to be consistent when it comes to units of time: getTimePassed is in seconds; sleepTime is not denoted; the text above Algorithm 1 mentions milliseconds, date_ns appears to be in nanoseconds. Also, avg_resource_usage array appears not to be initialized at all. Finally, sleeptime should have been 5 secs (according to text) but the pseudocode calculates it dynamically.

Figure 5 should have a scale on the time (x) axis as well.

The reviewer believes that Figure 6 suggests a linear increase in memory (if outliers are removed). Is this true?

It would be beneficial to know the overall size of data sent from the edge devices to the HLF nodes.

Why did the authors choose external chaincode and not the standard way?
Why, what and how did the authors change the HLF implementation?

Minor comment: please check that Fabric and Kubernetes are capitalized on all occurences.

Reviewer B

Does this paper represent a novel contribution to cryptocurrency or blockchain scholarship?

Yes

If you answered "yes" to the previous question, in one sentence, describe in your own words the novel contribution made by this paper:

ISSN 2379-5980 (online)
associated article DOI
10.5195/LEDGER.2022.230

The contribution is novel because it expands on previous work (in permissionless blockchains) but makes it more scalable and perhaps even more useful.

Is the research framed within its scholarly context and does the paper cite appropriate prior works?

Yes

Please assess the article's level of academic rigor.

Excellent (terms are well defined, proofs/derivations are included for theoretical work, statistical tests are included for empirical studies, etc.)

Please assess the article's quality of presentation.

Excellent (the motivation for the work is clear, the prose is fluid and correct grammar is used, the main ideas are communicated concisely, and highly-technical details are relegated to appendixes).

How does the quality of this paper compare to other papers in this field?

Top 20%

Please provide your free-form review for the author in this section.

Great paper! Good job and congrats on this innovative approach.

I think you spend too long explaining what Kubernetes is and how it works. I would assume the reader has knowledge about the basic building blocks of K8 and cut down the length of the paper. Keeping the explanation on Hyperledger is good as it provides a clear difference between permission and permissionless blockchains.

Comparing your paper to the previous work of "Implementation of smart contracts for blockchain based IoT applications,by Papadodimas" a few times as well.

Your paper is written much better in how it explains the permissionless way of collecting metrics. However where it falls short is explaining more "real-world" use case. The Papadodimas paper gives example of a data broker weather dapp, and while you do have an example somewhat at the end of the document when you are showing your experimental results, I feel adding a concrete real world example will make this paper shine.

When I read the Papadodimas paper, I feel I can go and build something right away, when I read your paper, I feel inspired about this innovation but I also wonder what is a good example of where this would shine in the real world that would make me want to implement this right away.

ISSN 2379-5980 (online)
associated article DOI
10.5195/LEDGER.2022.230

On Page 12, the scenarios you explain get a bit complicated to read, rewriting this section to be a bit more clear will help.

The diagram on page 9, while clear, is not easy to understand right away. Is there a better way to redraw this diagram that would make it clear to the reader what is happening and why.

1B. Author Responses

The paper is written very nicely, and it is structured in a good manner. The reviewer believes that the largest potential for improvement is the motivation and clear justification of the "why blockchain?" and "how is blockchain beneficial" questions:

- We thank the reviewer for his positive comments. We tried to address the reviewer's comments, giving detailed explanations and clarifications per comment. Please see our answers regarding the corresponding comments in the subsections below.

Resource monitoring is not the type of data where tamper-proofness, trustworthiness and other DLT core properties are of highest importance. What is the advantage of DLT-based monitoring compared to, for example, MoM-based data collection? With n nodes, the amount of raw monitoring data is stored n-fold; maybe it is more economical to be reduced from n to a smaller number, e.g. "max of 4"?

- Indeed, in this proof of concept scenario the system resources used may not appear to be of critical importance regarding their tamper-proofness. However, similar metrics, acquired from hardware performance counters, e.g., instructions per cycle (IPC), cache misses, branch predictions, may reveal sensitive information that can be exploited by malicious users that have access to the network. For example, prior scientific works have shown the existence of a correlation between hardware performance counters and execution latency of applications [R1, R2]. This information may allow them to infer the type or performance of workloads executed on a specific node, and to get insights regarding the vulnerabilities and the type of attack suitable for the current system state.
- The added value of the DLT-based monitoring, in this case, occurs also from the trust based on the single truth regarding resource utilization, stored in the shared ledger. The indestructible tracking of monitored resources can be beneficial for both the resource providers to be aware of the resource usage of their hardware, as well as the users, to evaluate the hardware they are utilizing and get billed for. In this concept, similar initiatives, i.e., the Pledger project1, have recognized the need for cross-layer knowledge between the service providers and the users, and explored the feasibility of such distributed ledger applications on the edge. Thus, an n-fold storing of the data (with the n depending on the use-case scenario, could guarantee the transparency among the different parties in the network.
- In the revised version of the manuscript (section 1 (Introduction), page 3) we have added the following sentences to summarize the above discussion:
- Resource monitoring is a field that can be explored as a distributed ledger application. Resource usage and micro-architectural events can be exploited by malicious users to infer the type of workload executed on the machine. Therefore such resource

ISSN 2379-5980 (online)
associated article DOI
10.5195/LEDGER.2022.230

information should be transferred to trusted members of a network, by leveraging the DLT. The added value of the DLT-based monitoring in this case, occurs also from the trust based on the single truth regarding resource utilization, stored in the shared ledger. The indestructible tracking of monitored resources, can be beneficial for both the resource providers to be aware of the resource usage of their hardware, as well as the users, to evaluate the hardware they are utilizing and get billed for. In this concept, similar initiatives, i.e., the Pledger project, have recognized the need for cross-layer knowledge between the service providers and the users, and explore the feasibility of such distributed ledger applications on the edge.

Why is consensus over raw monitoring data necessary at all? The consensus-participating edge nodes/organizations cannot verify the accuracy of the raw data because that data is only verifiable on the edge device itself.

- In the scenario discussed in the presented proof of concept, different organizations own the edge devices, which are part of the DLT network. As it was also mentioned in A.2, the vision of the proposed design is to create a network that will be able to verify the history of the conducted transactions of the blockchain and not the integrity of the monitoring data themselves. The verification of the transactions can evaluate the offered services by an organization to another in terms of resource availability, bandwidth and speed. This evaluation may be twofold. Resource providers can verify the volume, duration and allocation of the offered resources, while clients may benefit from verifying the provided resources to evaluate any Service-Level Agreements (SLAs).
- Of course, as the reviewer mentions, there could be cases that the reported monitoring data are inaccurate, either intentionally (e.g., man in the middle attacks) or unintentionally (e.g., malfunctioning devices). In such circumstances, the integrity of the raw data can be verified through well-defined smart contracts on the blockchain. In fact, recent scientific works have proposed consensus algorithms for verifying the monitoring data in distributed IoT environments [R3]. Such algorithms can be integrated within the DLT-based monitoring system to validate the transactions' information. However, this approach is out of the scope of this work, since it forms a completely different topic of research.
- In the revised version of the manuscript (section 1 (Introduction), page 3) we have added the following sentence to clarify the above.
- The added value of the DLT-based monitoring in this case, occurs also from the trust based on the single truth regarding resource utilization, stored in the shared ledger. The indestructible tracking of monitored resources, can be beneficial for both the resource providers to be aware of the resource usage of their hardware, as well as the users, to evaluate the hardware they are utilizing and get billed for.

Consensus-finding incurs overheads (especially during the coordination), how does that overhead relate to the claim that a DLT-based solution is avoiding the downsides of the

ISSN 2379-5980 (online)
associated article DOI
10.5195/LEDGER.2022.230

centralized data collection? (in the paper's abstract, the authors say that the centralized data collection approaches "are unable to handle the increased load")

- Our claim has not been evaluated experimentally in this paper. However, since centralized time series monitoring solutions such as Prometheus, and InfluxDB present various inefficiencies [R4], a comparison study could reveal useful performance comparisons over different aspects of the two approaches, e.g., scalability, latency, storage footprint. For example, different read/write ratios may result in the selection of a different approach as the most efficient. In scenarios in which reads are much more than the writes, peers would benefit from reading their local storage without unnecessary communications through the network. Regarding fault-tolerance a decentralized monitoring solution could offer increased reliability. On the other side, any unresponsiveness of the server node in the centralized approach could induce severe delays and service interruptions that cannot be tolerated.
- In the revised version, we have moved it to the future work section, in order to explore the cases in which the use of decentralized monitoring is more beneficial compared to centralized approaches.

The blockchain network is technically and organizationally decentralized, but the network is logically centralized as it has a shared single truth, completely replicated across nodes (with some delays, and network outages). How do edge devices (which are not only "HLF clients" but actual HLF nodes) choose which other HLF node(s) to gossip to?

- In a real-life deployment, the proposed solution may consist of various channels2, which form private "subnets" of communication between two or more specific network members. Each channel can be initialized with a predefined set of trusted partners and organizations. The single truth is decentralized in order to be accessed, reconfigured and maintained with the consent of every participating member.

How does a DLT-based solution compare (conceptually) to a load-balanced cluster of monitoring servers, with the monitoring subjects choosing a server randomly on each report?

- Conceptually, the load-balancing approach still has a single point of entry in the cluster. In addition, database synchronization delays may occur [R5]. Still, in a load balanced cluster, the integrity of the servers must be guaranteed and accepted by every member of the network. Therefore, the main qualitative difference is the voluntary absence of a central authority that serves the members of the network.

With unbounded ledger growth and no deletion facilities in HLF, how is the stored data rotated (i.e. discarded)?

- The volume of the extracted data of the examined scenario is low (about 25 bytes per reporting device). However, in cases with a greater data footprint, we can consider using an off-chain store, i.e., InterPlanetary File System (IPFS) [R6], in order to address the unbounded data growth.

ISSN 2379-5980 (online)
associated article DOI
10.5195/LEDGER.2022.230

Another important aspect which needs improvement is the case study with measurements. The reviewer believes that to go beyond 3 edge devices (which is way too few), a simulated (e.g. cloud-based) IoT network is useful (but also easy to set up, because the virtual devices can be bootstrapped based on the same template as for RasPi devices). Following the paper's 5-sec measurement interval, one should be able to see the effects of both spikes (many edge devices sending out data) and flats (average load). In particular, a larger number of edge devices would enable conclusions about throughput and latency, and not just about processing time and overheads.

- For the scope of this work, we focused more on the integration of Kubernetes with the HyperLedger Fabric on power-constrained devices, and the purpose of the presented evaluation was mainly for feasibility proofness. We are planning to extend the currently submitted work in the future, in order to explore the framework in a cluster beyond three homogeneous devices. However, in a larger, heterogeneous cluster, the overhead and the performance, may present variability.

Figure 4 needs a rework to clearly display which functionality is deployed on edge device and which is not.

- In our proposed solution every component and DLT functionality is deployed on edge devices exclusively. However, we redrew Fig. 4 in order to be more easily understandable for the reader. We separated the labels that describe each system flow step (on the right) from the DLT components and the flowchart (on the left).

Algorithm 1 (why not provide Golang chaincode source code, too?) appears not to be consistent when it comes to units of time: getTimePassed is in seconds; sleepTime is not denoted; the text above Algorithm 1 mentions milliseconds, date_ns appears to be in nanoseconds. Also, avg_resource_usage array appears not to be initialized at all. Finally, sleeptime should have been 5 secs (according to text) but the pseudocode calculates it dynamically.

- The Golang code is extensive (mostly boilerplate code) and in our opinion does not add any value to the comprehension of the manuscript. However, we can include it in an Appendix section if required by the reviewers.
- In the pseudocode, we calculate the sleep time depending on the time of the latest metrics acquisition. Thus, if metrics were monitored in t0, and stored in t1, we will set the sleep time to 5-(t1-t0). We selected this policy in order to be precise with the time that the metrics have been monitored from the system.
- In the revised version, we changed the runtime variable to seconds, the sleeptime to milliseconds granularity, and removed the avg_resource_usage array.

Figure 5 should have a scale on the time (x) axis as well.

- We thank the reviewer for the note. We have added the time scale (seconds) in the revised manuscript.

ISSN 2379-5980 (online)
associated article DOI
10.5195/LEDGER.2022.230

The reviewer believes that Figure 6 suggests a linear increase in memory (if outliers are removed). Is this true?

- Indeed, Figure 6 suggests an increase in the memory used in the devices. This occurs because of the database (LevelDB by default) that are the ledger blocks stored, which consumes a predefined amount of memory in order to cache the latest blocks inserted in the chain. In the present setup, this database is embedded in the peers. Therefore, the amount of the allocated memory over time increases until it reaches the limit defined during the network creation (64MB by default). If this limit is set to 0MB, the cache gets disabled. Thus the user can tune this value with the amount of memory used for caching, depending on the desired window in the ledger history that needs to be accessed with low latency.
- In the revised version of the manuscript (section 5.2, page 12) we have added a paragraph describing the discussion above.
- In both of the deployed scenarios, an increase in the utilized memory is illustrated in Fig. 6. This behavior is justified by the presence of an embedded in the peer database utilized by the HyperLedger Fabric framework. This database caches the latest blocks inserted in the ledger, based on a user-specified amount of memory for low latency retrieval. Thus, the user can tune this value with the amount of memory used for caching, depending on the desired window in the ledger history that needs to be accessed with low latency.

It would be beneficial to know the overall size of data sent from the edge devices to the HLF nodes.

- The overall size of every resource usage report is ~25 bytes.

Why did the authors choose external chaincode and not the standard way?

- The external chaincode allows chaincode runtime management independently of the peer. This design decision runs the chaincode as a service, which applies to the cloud-native principles of Kubernetes. Additionally, since the HyperLedger Fabric components are dockerized for simplified management and deployment, building and executing the dockerized chaincode inside another docker container (Docker in Docker) is a design method that should be avoided.

Why, what and how did the authors change the HLF implementation?

- In this paper, we presented the integration of Kubernetes and HyperLedger Fabric to a cluster of power-constrained, edge devices. In order to adapt to Kubernetes primitives, we followed the containerized approach for the deployment of HLF's building blocks. The changes in the HLF implementation are limited to the re-building of every stage of the various components, i.e., peer, orderer, etc., to enable ARM processor compatibility. The provided, open-sourced images are compatible with x86 only processors. Thus, we rebuilt the docker images using the multi-arch build feature.

ISSN 2379-5980 (online)
associated article DOI
10.5195/LEDGER.2022.230

Reviewer B

Great paper! Good job and congrats on this innovative approach.

- The authors would like to thank the reviewer for his encouraging comments and feedback provided.

I think you spend too long explaining what Kubernetes is and how it works. I would assume the reader has knowledge about the basic building blocks of K8s and cut down the length of the paper. Keeping the explanation on Hyperledger is good as it provides a clear difference between permission and permissionless blockchains.

- The authors assumed that since the topic of the journal is more related to blockchain technologies, a brief description of Kubernetes' basic building blocks would be beneficial for facilitating the readers' overall understanding.
- In the revised version of the manuscript (section 3.1, pages 5-6) we have reduced the description of the Kubernetes' basic building blocks and features. More precisely, we have merged the descriptions of Pod, Deployment, and Service into a single paragraph, and removed the description of the Affinity feature.

Comparing your paper to the previous work of "Implementation of smart contracts for blockchain based IoT applications, by Papadodimas" a few times as well. Your paper is written much better in how it explains the permissionless way of collecting metrics.

- We would like to thank the reviewer again for his positive comments.

However where it falls short is explaining more "real-world" use case. The Papadodimas paper gives example of a data broker weather dapp, and while you do have an example somewhat at the end of the document when you are showing your experimental results, I feel adding a concrete real world example will make this paper shine.

When I read the Papadodimas paper, I feel I can go and build something right away, when I read your paper, I feel inspired about this innovation but I also wonder what is a good example of where this would shine in the real world that would make me want to implement this right away.

- We understand the comment of the reviewer at this point. The paper of Papadodimas focuses more on the implementation of a real-world decentralized application that can be designed and used by the reader. However, the purpose of our work was mostly to motivate through a proof of concept implementation the exploitation of edge, power-constrained devices with limited resources, for designing a blockchain network for monitoring purposes. We demonstrate a decentralized monitoring example to verify the viability of such a use-case. More tangible, real-world applications of this PoC is the work conducted under the Pledger project3, in which the need for cross-layer knowledge between the service providers and the users is recognized, and the feasibility of such distributed ledger applications on the edge is explored. More

ISSN 2379-5980 (online)
associated article DOI
10.5195/LEDGER.2022.230

specifically, this design benefits both the hardware resource providers that require improved awareness on the types of applications executed by their customers, and the consumers of the infrastructure that require improved awareness on the types of physical nodes their applications will be executed on.

On Page 12, the scenarios you explain get a bit complicated to read, rewriting this section to be a bit more clear will help.

- In this subsection we describe and measure two different scenarios. In the first one, both organizations/agents monitor and store periodically the resource usage metrics in the ledger. Therefore both participating organizations conduct both store and load operations on the shared, distributed ledger at the same time. However, in the second scenario, we monitor the device's resource utilization when organization no. 2 queries the shared ledger. This scenario describes the case in which the role of a participating member in the blockchain network is restricted to accessing the ledger for information retrieval. Thus, by executing this scenario, we demonstrate the cost of querying the ledger entries, in terms of resource usage and latency.
- In the revised version of the manuscript (section 5.2, pages 11-12) we have explained the different scenarios setups, by adding a more precise description where it was necessary.

The diagram on page 9, while clear, is not easy to understand right away. Is there a better way to redraw this diagram that would make it clear to the reader what is happening and why.

- We redrew the diagram and separated the labels with the system flow description (on the right), from the DLT flowchart (on the left).

Revision Letter References

[R1] Jia, Z., Zhan, J., Wang, L., Luo, C., Gao, W., Jin, Y., ... & Zhang, L. (2016). Understanding big data analytics workloads on modern processors. IEEE Transactions on Parallel and Distributed Systems, 28(6), 1797-1810.

[R2] Masouros, D., Xydis, S., & Soudris, D. (2020). Rusty: Runtime interference-aware predictive monitoring for modern multi-tenant systems. IEEE Transactions on Parallel and Distributed Systems, 32(1), 184-198.

[R3] Aldana, J. A. A., Maag, S., & Zaidi, F. (2021). A formal consensus-based distributed monitoring approach for mobile IoT networks. Internet of Things, 13, 100352.

[R4] Wang, Z., Xue, J., & Shao, Z. (2021). Heracles: an efficient storage model and data flushing for performance monitoring timeseries. Proceedings of the VLDB Endowment, 14(6), 1080-1092.

[R5] Faleiro, Jose M., and Daniel J. Abadi. "Latch-free synchronization in database systems: Silver bullet or fool's gold?." CIDR (Conference on Innovative Data Systems Research). 2017.

[R6] IPFS, https://ipfs.io/, 2022.

ISSN 2379-5980 (online)
associated article DOI
10.5195/LEDGER.2022.230

2A. Second Round Review

Reviewer A

Did you review an earlier version of this submission? (If "no," please contact the editor.)

Yes

Has the submission been sufficiently revised to address your previous concerns?

Yes

Do you have any new concerns specific to this revision?

Yes

If you answered "yes" to the previous question, please provide more detailed feedback here.

The improvements are visible, although a few "low-hanging fruits" need to be picked from the tree.

In Figure 4, the presences/absence of arrowheads, as well as the thickness/dashing of arrows/lines is not explained. For some of the actions (e.g. #6), the circles are located on the participating roles (e.g. peers), not on arrows/lines. This *must* be improved, although A.9 says that the picture was already redrawn.

As the IPFS is mentioned in the comments to the editor, I would also suggest to mention it in the paper itself. This is not yet the case, but the discussion of data/ledger growth and deletion facilities should be reflected in the paper, not just in the comments to the editor.

The authors should include a link to Pledger (a link is included in the comments to the editor, but not in the paper itself).

In the references: "Edge Computing: Vision and Challenges": both the paper title and the journal title are not accurate regarding capitalization. The same is true for the KEIDS paper of Kaur et al.; all other references should also be carefully checked for capitalization. Also, some references are incomplete (e.g. #31, with year etc. missing).

ISSN 2379-5980 (online)
associated article DOI
10.5195/LEDGER.2022.230