Blockchain for Trustworthy Publication and Integration of Linked Open Data

Fabian Kirstein
fabian.kirstein@fokus.fraunhofer.de
Fraunhofer FOKUS
Berlin, Germany
Weizenbaum Institute
Berlin, Germany

Manfred Hauswirth
manfred.hauswirth@tu-berlin.de
TU Berlin, Open Distributed Systems
Berlin, Germany
Weizenbaum Institute
Berlin, Germany

ABSTRACT

The timely, traceable and provenance-aware publication of Linked Open Data (LOD) is crucial for its success and to fulfill the vision of a global, decentralized, and machine-readable database of knowledge. Yet, the access to LOD is still fragmented and mainly centralized aggregations are being used, relying on complex harvesting mechanisms. As a remedy, we propose a blockchain-based approach enabling an integrated, traceable, and timely view on LOD. We use a blockchain to meet the organizational requirements of publishing LOD in a decentralized fashion while still supporting the sovereignty of the data providers and supporting provenance and proper integration into a harmonized knowledge graph. We present an approach and an implemented system that fulfills the requirements regarding volume and throughput and can be used as the foundation for practical deployments. We use Linked Open Government Data (LOGD) as our case study to demonstrate the feasibility of our approach. We developed a prototype to address the specific requirements of LOGD publication and apply the Practical Byzantine Fault Tolerance algorithm at its core to enable a robust state replication.

CCS CONCEPTS

• **Information systems** → **Information integration**; • **Computer systems organization** → **Peer-to-peer architectures**.

KEYWORDS

Linked Open Data; Blockchain; PBFT; DCAT

ACM Reference Format:
Fabian Kirstein and Manfred Hauswirth. 2021. Blockchain for Trustworthy Publication and Integration of Linked Open Data. In *Proceedings of the 11th Knowledge Capture Conference (K-CAP '21), December 2–3, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3460210.3493572

1 INTRODUCTION

Seemingly "centralized" access, while underneath being a truly decentralized distributed information system with complete control of the data by the data providers, is a key architectural principle of modern Linked Open Data (LOD) infrastructures. From a conceptual point of view, this may seem optimal, but it comes at the prize of significantly higher algorithmic complexity and negative impact on the system performance and general drawbacks, like low data quality, weak interoperability, unreliable availability, and little encouragement for community involvement [7].

Blockchains could help to mitigate some of these problems. For example, in the healthcare domain [10] or logistics [12] blockchains are being researched actively for their potential to address key problems in those domains. In this paper, we investigate if blockchains could be used to address a set of key problems in the publication of Linked Data with the specific example of Linked Open Government Data (LOGD): (1) Distributed dissemination and publication implies time-delayed and pull-based harvesting infrastructures, like data.europa.eu (DEU)[1] or Google Dataset Search [8]. Here, a blockchain could act as a virtual access point, managed as a shared common state between multiple data providers. In addition, the onboarding process for new data providers can be accelerated. (2) LOGD is known to be challenged by low data quality and heterogeneous data structures and interfaces. The consensus and immutability in a blockchain can support the assertion of homogeneous data quality and format standards, and force data publishers to apply a thorough quality assurance process. (3) In most cases, the current publication scheme does not provide provenance or trust mechanisms. The blockchain's immutability can be used as a provenance layer and store the history of individual datasets. (4) LOGD publication is a closed process with high-level stakeholders involved, where third-party or community involvement is not stipulated. More decentralization through blockchain can support the democratization of the entire LOGD life cycle by flattening hierarchies, increasing transparency, and laying the foundation for opening up infrastructures to participants beyond established stakeholders.

In this paper we present our initial results to create a decentralized and distributed network for publishing and accessing LOGD, which complies with established publication processes and methodologies for Open Data and which is based on the application of blockchain techniques to provide provenance and trust mechanisms.

2 BACKGROUND AND RELATED WORK

Open Data is about the open provisioning of (structured) datasets and knowledge on the Web [2] and is known as an important source

[1] https://data.europa.eu

K-CAP '21, December 2–3, 2021, Virtual Event, USA.

ACM ISBN 978-1-4503-8457-5/21/12.
https://doi.org/10.1145/3460210.3493572

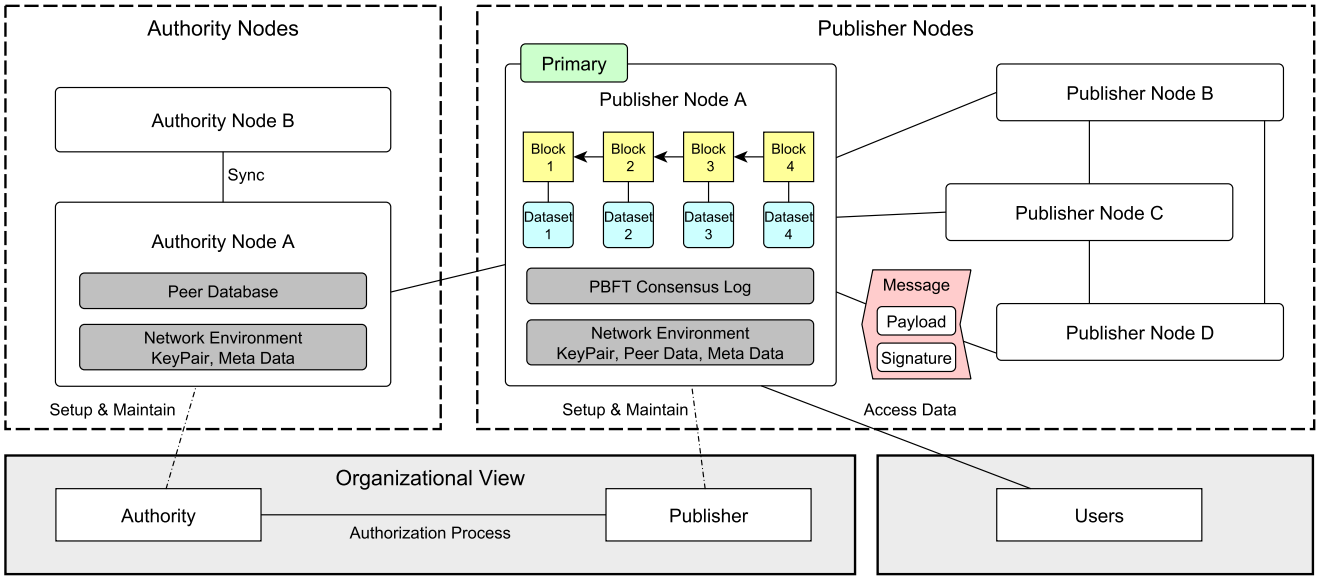

Figure 1: Overview of the Open Data Blockchain network. Publisher Node A and Authority Node A are depicted in more detail.

for LOGD [9]. In Europe, it is published in a bottom-up fashion and its providers form a graph, where national providers gather data from regional providers. This bottom-up approach of collecting data from the lowest level to a central portal defines the availability of the datasets in all intermediate portals. Similar hierarchies can be found in related domains. For instance, scientific publications are released via institutional repositories, aggregated in national research portals, and further published in central portals. LOGD incurs a high degree of offline communication and management between the stakeholders (mainly public authorities). We performed an analysis of the current data pool of data.europa.eu[2] to determine its characteristics regarding volume, velocity, and variety. We found that the current graph has a size of ~20GB, each dataset is around 20kB, the update frequency is approx. 10 datasets / minute and the Data Catalogue Vocabulary (DCAT) specification is employed as data model. The 30 biggest data providers account for close to 98% of all available datasets.

Some initial approaches exist on the combination of blockchain technologies and Linked Open Data (LOD). English et al. propose to improve the persistent identification of Resource Description Framework (RDF) resources with blockchain [3]. Third et al. theoretically investigate stages of managing Linked Data in a distributed ledger, from a verification layer to a full storage layer, and propose integrating the concepts into existing data architectures [11]. Other work exists about the combination of blockchain and Open Data, e.g., Truong et al. [14] developed a portal based on Hyperledger Fabric to increase the availability and integrity of open datasets. However, they only consider a single data provider and do not incorporate the network and stakeholder structure. Regerator [13] is a solution to publish Open Data on the Ethereum blockchain via a smart contract to offer integrity, immutability, and availability. It depends on the public Ethereum network, causing transactions costs for every published dataset. To the best of our knowledge, there is no specific blockchain-based solution, which is tailored to the current state and methodology of LOGD and its actors.

A blockchain is an immutable, distributed data store, which is structured as a list of blocks, where each block is cryptographically linked to its predecessor [17]. Since its introduction with Bitcoin [6] the concepts of decentralization, immutability, and traceability have inspired a plethora of use cases and similar solutions [18]. A fundamental challenge of a blockchain is that all peers need to achieve a common and consistent view of the state in a given timeframe and tolerate faults of individual peers. This problem is addressed by a so-called consensus protocol. A lot of research was conducted in this field, one of the most significant contributions is the Paxos protocol by Leslie Lamport [5]. It enables reaching consensus in an unreliable network by exchanging a set of well-defined messages in a sequence of communication rounds. Subsequent research improved the protocol and introduced alternative approaches, particularly in the blockchain domain. Interested readers are referred to the extensive survey by Xiao et al. [16]. Some of the protocols are inspired by the Practical Byzantine Fault Tolerance (PBFT) protocol by Castro and Liskov [1]. PBFT is a high-performance state machine replication algorithm to tolerate Byzantine faults in asynchronous systems. It provides safety and liveness if at most $\lfloor \frac{n-1}{3} \rfloor$ out of n nodes are faulty and has proven to be reliable and robust.

3 DESIGN AND IMPLEMENTATION

We identified a set of key requirements for LOGD, guiding the design and implementation of our system: (1) The metadata of a LOGD ecosystem is distributed and decentralized across multiple stakeholders and exchanged among them. The governance model is highly decentralized, with no stakeholder controlling the network.

[2]The official portal for European data - https://data.europa.eu.

(2) The data is accessible openly without permission from any node in the network. The data should always be presented consistently throughout the entire network irrespective of the point of access. (3) The write access to the network is governed by multiple authorities. (4) The authorization and authentication of publishers is based on known offline identities and maintained via established communication channels between the stakeholders. (5) The network supports the already existing volume and velocity of a typical LOGD network and is scalable beyond that.

We decided to implement a custom blockchain for a first evaluation. We found that existing blockchain frameworks are either too narrow in their potential applications (operating cryptocurrencies) or are too extensive and complex for our use case. We call our approach **ODBl** (Open Data Blockchain) and implemented it as a working prototype based on Vert.x and MongoDB.[3] We evaluated it with a set of datasets, which meet the real-world requirements of LOGD to test the applicability and performance. Figure 1 provides an overview of the ODBl architecture.

Our solution employs a two-level hierarchy. The two levels are represented as different types of nodes: the publisher nodes P and the authority nodes A. Publisher nodes denote the role of a publisher (on all levels), hence such nodes act as LOGD portals and are being operated accordingly. Authority nodes grant write access and support the bootstrap process for new nodes. ODBl is a "consortium blockchain" [18] with full replication among the publisher nodes. The write access to the network must be granted and revoked explicitly for every node. The initial startup for a publisher node is a three-step process: (1) bootstrap and configure a node, (2) register and authenticate the node with an authority node, and (3) join the network and synchronize with the latest state. For robust state replication and consensus across all publisher nodes, we applied the PBFT algorithm [1]. It provides sufficient safety and liveness for our network. Consequently, ODBl applies a voting-based consensus mechanism, where a new block is added if $\lceil \frac{2}{3}n + 1 \rceil$ nodes validate it as correct. A new transaction (issued by a request) indicates the intention to append a new dataset to the data pool of the network or modify or delete an existing dataset. The well-established DCAT specification is used as the representation format for the datasets [15]. We provide a mechanism for appending new versions to an existing dataset, by providing a state indicator within the transaction. This mechanism allows the system to maintain basic provenance information within the network since it creates an immutable version history of the datasets. We apply well-established cryptography approaches (RSA-PKI with at least 2048 bit and SHA-256 hashes) to implement access control and to sign the blockchain data structure.

4 RESULTS AND OUTLOOK

We created a simulated environment (on a Kubernetes cluster) to evaluate the functionality and performance of ODBl under real-world conditions. We deployed 30 publisher nodes and one authority node to emulate the structure of data.europa.eu. We scripted several operations and use cases for the network and measured CPU load and memory consumption. Each node was configured with a synthetic, realistic network delay. We measured the performance

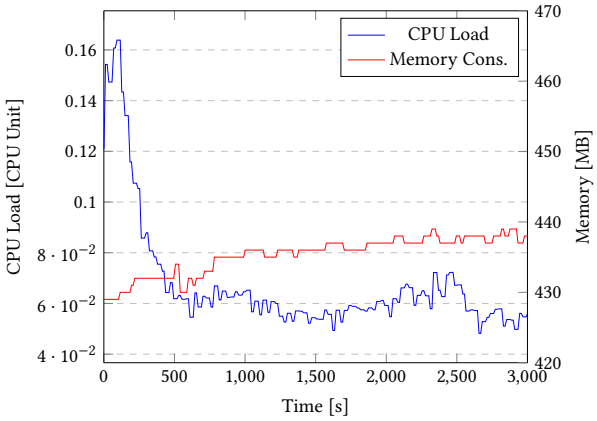

Figure 2: CPU and Memory Load during Dataset Publication

of a single node using the built-in monitoring tool of Kubernetes. Memory usage is measured in megabytes (MB) and CPU load in CPU units. A CPU unit corresponds to the workload of a single core in the underlying processor (Intel Xeon Gold Prozessor 6126 (12 x 2.60 GHz)). The detailed tests are described in the following.[4] **(1) Initial Setup:** We deployed the authority node and gradually bootstrapped the publisher nodes. Each publisher node performed the authentication process with the authority node, including a simulated public key exchange. The process was concluded after the information from all nodes was distributed successfully in the network, which took 303 seconds. **(2) Dataset Publication:** We simulated the normal operation of the network by constantly publishing datasets across all nodes. We issued around 13 datasets per minute for 1 hour, totaling 815 datasets. This corresponds to the typical real-world update frequency, discussed in Section 2. The maximum CPU load was 0.16 CPU units, the average CPU load was 0.06 CPU units, which is only a very small fraction of the available resources. The maximum memory consumption was 439 MB and the average memory consumption was 436 MB, for a single node, which can be considered as very moderate. Figure 2 presents the resource consumption during the test. The memory consumption is close to constant, whereas the CPU load has a high peak at the beginning of the process, caused by general initialization of the application, e.g., reopening database connections.

(3) Faulty Nodes and Failing Primary: We deliberately created faulty nodes by shutting down nodes up to the maximum supported number of faulty nodes of $\frac{1}{3}n - 1$, i.e., 9 of our 30 nodes. By this, we tested if the non-faulty nodes could maintain full operation, if the faulty nodes successfully synchronized with the up-to-date state after reactivation, and if the network can deal with a failing primary. While continuing to issue new datasets, we measured the time the network required to recover completely. In all cases, the network continued to operate and reached consensus successfully in a reasonable time. The recovery time was always between 70 and 150 seconds and does not correlate with the number of faulty nodes. **(4) Client Access:** We simulated load on a single node interface to

[3] ODBl is available as open source: https://github.com/odbl/odbl

[4] All test scripts and results can be found here: https://github.com/odbl/odbl-evaluation

evaluate if a node can successfully deal with load profiles of realistic production environments. As a preparation for this evaluation, we inserted 500.000 real datasets into a node to approximate the real-world volume of LOGD. We simulated the parallel access load of 150 users per hour, where each user was executing 6 requests, i.e., a total of 900 requests per hour. These numbers were based on actual usage statistics of data.europa.eu [4]. The node responded flawlessly with an average response time of 700 ms.

The timely, traceable and provenance-aware publication of LOD is crucial for its success. Subsequent (third-party) tasks, like data analysis and reasoning, depend on a reliable and trustworthy knowledge base. The publication process of LOD is intrinsically decentralized, leading to high complexity. Yet, this provides us with many opportunities to shape the technical solutions according to the actual requirements and organizational environments of specific LOD providers. Our approach addresses the organizational and technical aspects of the LOGD domain. The required communication and governance channels between the stakeholders are already established and can be leveraged for the network. ODBl provides many advantages to publish and disseminate LOGD with a similar effort to existing approaches but provides additional advantages. Table 1 presents a comparison between our work and the established aggregation/harvesting approach in LOGD.

	Traditional LOGD	**ODBl**
Pros	Less redundancy Simpler to implement Central (quality) control	Timely distribution Single view on the data Improved quality through consensus Provenance and integrity through immutability Robustness More openness and transparency
Cons	Delayed aggregation Favors fragmentation Promotes proprietary centralization Quality compliance impeded No harmonized provenance and integrity tracking Single point of failure	Redundancy and communication overhead Requires a minimal number of participants Protocol updates need to be coordinated

Table 1: Comparison of traditional LOGD and ODBl

Our prototype offers a synchronized and consistent view on the data. This single point of truth is augmented with blockchain-based provenance. The consensus mechanism and immutability facilitate strict compliance with quality specifications. ODBl supports the goals of LOD by promoting independence from centralized aggregation services, enabling true sovereignty of the data publishers, and avoiding a de-facto monopoly situation of a single entity. However, the adoption of our approach is still challenging. An immutable state stored decentrally is partly disruptive to established methods. It creates a certain degree of redundancy and requires the rethinking of the software deployment/updates process. The practical success of ODBl relies on a network effect and would need to be initiated by a group of at least four publishers and an authority.

In the future, we want to further improve the functionality of ODBl and investigate the application of existing blockchain frameworks. This includes performance optimizations to deal with even larger amounts of data, e.g., through compression, increasing transaction size and more effective storage of dataset revisions. A special area of interest will be the investigation of third-party contributions to foster the involvement of independent data transformers and the broader LOD and knowledge management community.

ACKNOWLEDGMENTS

This work has been funded by the Federal Ministry of Education and Research of Germany (BMBF) under grant no. 16DII128 ("Deutsches Internet-Institut").

REFERENCES

[1] Miguel Castro and Barbara Liskov. 1999. Practical Byzantine fault tolerance. (1999), 173–186.
[2] Yannis Charalabidis, Anneke Zuiderwijk, Charalampos Alexopoulos, Marijn Janssen, Thomas Lampoltshammer, and Enrico Ferro. 2018. The Open Data Landscape. In *The World of Open Data: Concepts, Methods, Tools and Experiences*, Yannis Charalabidis, Anneke Zuiderwijk, Charalampos Alexopoulos, Marijn Janssen, Thomas Lampoltshammer, and Enrico Ferro (Eds.). Springer International Publishing, Cham, 1–9. https://doi.org/10.1007/978-3-319-90850-2_1
[3] Matthew English, Soren Auer, and John Domingue. [n. d.]. Block Chain Technologies & The Semantic Web: A Framework for Symbiotic Development. *The Semantic Web* ([n. d.]), 15.
[4] Fabian Kirstein, Benjamin Dittwald, Simon Dutkowski, Yury Glikman, Sonja Schimmler, and Manfred Hauswirth. 2019. Linked Data in the European Data Portal: A Comprehensive Platform for Applying DCAT-AP. In *EGOV2019*.
[5] Leslie Lamport. 1998. The Part-Time Parliament. *ACM Transactions on Computer Systems* 16, 2 (May 1998), 133–169. https://doi.org/10.1145/279227.279229
[6] Satoshi Nakamoto. 2009. Bitcoin : A Peer-to-Peer Electronic Cash System. (2009).
[7] Sebastian Neumaier, Lörinc Thurnay, Thomas J. Lampoltshammer, and Tomá Knap. 2018. Search, Filter, Fork, and Link Open Data: The ADEQUATe Platform: Data- and Community-Driven Quality Improvements *(WWW '18)*.
[8] Natasha Noy, Matthew Burgess, and Dan Brickley. 2019. Google Dataset Search: Building a Search Engine for Datasets in an Open Web Ecosystem. In *WebConf 2019)*.
[9] N. Shadbolt, K. O'Hara, T. Berners-Lee, N. Gibbins, H. Glaser, W. Hall, and m c schraefel. 2012. Linked Open Government Data: Lessons from Data.Gov.Uk. *IEEE Intelligent Systems* 27, 3 (2012).
[10] Asad Ali Siyal, Aisha Zahid Junejo, Muhammad Zawish, Kainat Ahmed, Aiman Khalil, and Georgia Soursou. 2019. Applications of Blockchain Technology in Medicine and Healthcare: Challenges and Future Perspectives. *Cryptography* 3, 1 (March 2019), 3. https://doi.org/10.3390/cryptography3010003
[11] Allan Third and John Domingue. [n. d.]. LinkChains: Exploring the Space of Decentralised Trustworthy Linked Data. ([n. d.]), 8.
[12] Edvard Tijan, Saša Aksentijević, Katarina Ivanić, and Mladen Jardas. 2019. Blockchain Technology Implementation in Logistics. *Sustainability* 11, 4 (Jan. 2019), 1185. https://doi.org/10.3390/su11041185
[13] A. Tran, X. Xu, I. Weber, M. Staples, and Paul Rimba. 2017. Regerator: a Registry Generator for Blockchain. (2017).
[14] Dinh-Duc Truong, Thanh Nguyen-Van, Quoc-Bao Nguyen, Nguyen Huynh Huy, Tuan-Anh Tran, Nhat-Quang Le, and Khuong Nguyen-An. 2019. Blockchain-Based Open Data: An Approach for Resolving Data Integrity and Transparency. In *Future Data and Security Engineering*. Vol. 11814. Springer.
[15] W3C. 2020. Data Catalog Vocabulary (DCAT). https://www.w3.org/TR/vocab-dcat/.
[16] Yang Xiao, Ning Zhang, Wenjing Lou, and Y. Thomas Hou. Secondquarter 2020. A Survey of Distributed Consensus Protocols for Blockchain Networks. *IEEE Communications Surveys Tutorials* 22, 2 (Secondquarter 2020), 1432–1465. https://doi.org/10.1109/COMST.2020.2969706
[17] Xiwei Xu, Ingo Weber, and Mark Staples. 2019. Introduction. In *Architecture for Blockchain Applications*, Xiwei Xu, Ingo Weber, and Mark Staples (Eds.). Springer International Publishing, Cham, 3–25. https://doi.org/10.1007/978-3-030-03035-3_1
[18] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. 2018. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services* 14, 4 (2018), 352–375.