2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

BLOCKCHAIN-BASED SECURE ARCHIVING OF SENSOR MALFUNCTIONS IN STRUCTURAL HEALTH MONITORING SYSTEMS

Patricia Peralta[1], Thamer Al-Zuriqat[1], Benjamin Burse[2], Heinrich Söbke[3,4], and Kay Smarsly[1]
[1]Institute of Digital and Autonomous Construction, Hamburg University of Technology, Germany
[2]Chair of Software Engineering, Bauhaus University Weimar, Germany
[3]Chair of Interactive Digital Systems, University of Applied Science Weserbergland, Hameln, Germany
[4]Chair of Resource Management, Bauhaus University Weimar, Germany

Abstract

Structural health monitoring (SHM) systems are widely used to assess civil infrastructure, but sensor malfunctions may result in erroneous assessments. Secure archiving of sensor malfunctions is needed for understanding malfunction frequencies and causes. This paper presents a blockchain-based system for online secure archiving of sensor malfunctions, designed as a modular add-on for SHM systems. Using a public blockchain framework, the blockchain-based archiving system is implemented and validated for two use cases common in SHM, (i) faulty data and (ii) sensors replacements. The validation results demonstrate the potential of blockchain technology to securely archive sensor malfunctions in SHM systems online.

Introduction

As civil infrastructure continues to deteriorate due to aging, the increasing frequency of natural hazards driven by climate change, and intensified traffic loads, the demand for effective maintenance strategies grows. Structural health monitoring (SHM) has emerged as a vital tool in infrastructure maintenance, enabling the assessment of structural conditions through sensors deployed to monitor changes in the infrastructure (Law et al., 2014). SHM helps improve cost efficiency in infrastructure maintenance by complementing inspections activities towards early damage detection. SHM systems are typically composed of sensor nodes, whose proper operation impacts the overall performance and reliability of the SHM systems (Al-Zuriqat et al., 2023).

Sensor nodes in SHM systems may experience malfunctions, such as sensor faults and malicious nodes, which may lead to erroneous assessments, additional costs, or even data loss. On the one hand, hardware or software malfunctions, harsh environmental conditions, or signal interferences may cause sensor faults (Smarsly et al., 2014). Sensor nodes may exhibit single or combined sensor faults, such as complete failure, gain, bias, noise, drift, and outliers, hindering the accuracy of the SHM systems (Al-Nasser et al., 2024). On the other hand, malicious attacks may compromise sensor nodes into acting as malicious nodes (Ramasamy et al., 2021), which is particularly concerning when monitoring critical infrastructure. When a sensor node presents security vulnerabilities that disrupt the operation of an SHM system, the sensor node may be considered a “malicious node” (Lai et al., 2022).

Diagnosing sensor faults and detecting malicious nodes in SHM systems allows to take actions intended to accommodate the sensors faults and to contain malicious nodes, mitigating possible errors. The process of fault diagnosis includes detecting, isolating, identifying, and accommodating sensor faults based on physical redundancy, analytical redundancy, or a combination of both (Bartels et al., 2024). Recent developments in fault diagnosis have been overviewed by Deng et al. (2024), where advantages and disadvantages of various methods based on analytical redundancy are discussed. Efforts have been explored for onboard autonomous fault detection facilitated by embedding computing into sensor nodes (Dragos et al., 2016). Detecting malicious nodes and tracking sensor malfunctions is essential for wireless SHM systems to prevent malicious attacks. Lai et al. (2022) have analyzed current methods for detecting malicious nodes, where the node behavior is evaluated under malicious attacks, including blackhole, Sybil, false data injection, random poisoning, and sinkhole attacks. Furthermore, it is necessary to track and analyze sensor malfunctions to gaining insight into malfunction frequencies and causes, as well as to gather domain-specific knowledge for advance fault diagnosis (Xie et al., 2024). However, proper tracking of sensor malfunctions has not been addressed yet. To track sensor malfunctions, a robust and secure archiving solution is required, offering tamper-proof capabilities to assess the vulnerability of SHM systems to sensor faults and malicious nodes, while enabling the generation of audit-compliant documentation.

SHM systems commonly utilize centralized data environments for data management and archiving. In centralized data environments, such as centralized servers and digital twins, data is stored in databases and provided for sensor data analyses (Brötzmann et al., 2022). One advantage of centralized data environments for SHM systems is the ability to facilitate tracking of sensor faults or malicious nodes by analyzing the sensor data (Moridi,

et al. 2020). However, centralized data environments present several drawbacks (Chen et al., 2024; Brötzmann et al., 2022), including

(i) the effort-intensive and time-consuming task of setting up and scaling databases for complex data,
(ii) the need for specialized knowledge to enable seamless data exchange between systems,
(iii) susceptibility to security risks and malicious attacks, and
(iv) the risk of complete system failure if the centralized environment experiences a malfunction, which could disrupt the online tracking and archiving of sensor faults and of malicious nodes.

Addressing the drawbacks of central data environments, blockchain technology is a promising approach for tamper-proof archiving, enhancing tracking of sensor malfunctions with online functionalities.

Decentralized data management solutions for sensor systems have been proposed using blockchain (Brötzmann et al., 2022) and edge computing for on-board data processing (Yu et al., 2024). Blockchains is a distributed ledger technology that organizes sequences of blocks to document transactions via cryptographic authentication (Crosby et al., 2016). Within blockchains, sensor data exchanges can be stored as tamper-proof and easily verifiable transactions, facilitating data storage and management of heterogenous data sources (Zhang et al., 2024) as well as malicious nodes detection (Ramasamy et al., 2021). Moreover, coupling of web technologies and blockchain for data management in SHM applications is a new research area (Gigli et al., 2022). Extending the applications of blockchain technology towards SHM systems with online functionalities, this paper presented the development of a blockchain-based archiving system aiming at secure online archiving of sensor malfunctions in SHM systems. The blockchain-based archiving system is realized using a public blockchain framework, validated for two use cases common to SHM, (i) faulty data and (ii) sensors replacements.

The reminder of this paper is structured as follows. First, the design and implementation of a blockchain-based archiving system for secure archiving of sensor malfunctions is presented. Second, the blockchain-based archiving system is validated to test functionality and performance. Finally, the work presented herein is summarized, and relevant conclusions are drawn.

Blockchain-based archiving of sensor malfunctions

The blockchain-based archiving system aims at a temper-proof, online storage of (i) data recorded from sensor nodes of SHM systems and (ii) data describing replacements of sensors in sensor nodes. It should be noted, while the latter is technically “metadata”, for the sake of simplicity and consistency, it will be referred to as “data” throughout the remainder of this paper. In this section, the design and implementation of the blockchain-based archiving system is described. First, the requirements are defined based on two typical SHM use cases, serving as basis for the system design system. Then, the system design and implementation, using the public blockchain platform *Ardor* (Jelurida Swiss SA, 2024), is presented.

Sensor malfunction use cases in SHM systems

The requirements for designing the blockchain-based archiving system are defined based on two generally valid, typical SHM use cases, (i) faulty data and (ii) sensors replacements. Both use cases require to store data tamper-proof in the blockchain and to provide the data for subsequent analysis via queries. Only authenticated trustworthy data is stored in the blockchain. The uses cases are described as follows:

- ***Faulty data* (*use case* 1):** Sensor nodes of modern SHM systems present edge-computing capabilities that can execute fault diagnosis algorithms, including automated fault detection and isolation (Fritz et al., 2022). During operation of a sensor node, alerts may be triggered when detecting faulty data. The alerts are automatically recorded by the sensor nodes (Steiner et al. 2019). Along with the alerts, the corresponding measurement data should be recorded to evaluate the fault diagnosis in audit processes to identify false positives.
- ***Sensor replacements* (*use case* 2):** For audit-compliant documentation of the sensor node components, information of the sensors installed in a sensor node is needed. When a sensor is replaced, both the sensor removed and the sensor installed must be documented, usually stating the sensor type and serial number. As opposed to use case 1, the documentation of sensor replacements is recorded manually, typically using personalized handheld devices (Ding et al., 2016).

Furthermore, the following security features in the sensor nodes and personalized handheld devices are considered to guarantee trustworthiness for both use cases,

- certificates are needed to install or update software applications,
- access control via user authentication using user credentials or biometric processes, and
- data transmission using secure communication protocols and data encryption.

Design and implementation

Based on the use cases, the structure of the sensor node data to be stored in the blockchain is described in a data model (Figure 1). As can be seen from Figure 1, the class *SensorNodeData* describes general parameters of a sensor node and is the superclass of two classes, *ComponentRemoval* (related to use case 2) and *SensorEvent* (related to use case 1), which specify the data generated by specific events. The class *ComponentRemoval* describes the data recorded when a

sensor is removed and documented, e.g. by personalized handheld devices. The class *SensorEvent* describes specific events that may originate from a *SensorNode*. The *SensorEvent* class is superclass of three classes:

- *HeartBeatData* describes the operational data periodically generated by a sensor node. Storing the operational data in a blockchain provides periodic proof of the sensor node functionality.
- *AlertData* describes specific states of a sensor node. Relevant states include faulty data states, which are diagnosed by the sensor node itself, such as drifts of the values recorded by the sensor node with regard to a predefined threshold.
- *MeasurementData* describes a set of values recorded by a sensor node at a specific point in time. The blockchain-based archiving system is not intended to continuously store the accumulating measurement data of a sensor node but is capable of storing sets of measurement values at selected points in time for audit-compliant documentation. Accordingly, the class *MeasurementData* holds a *Measurement* set.

The enumeration data types included in the semantic model are described in Table 1. The enumeration data types indicate the application field of the blockchain-based archiving system and reflect the extensibility of the data model.

The *Ardor* blockchain framework is selected for its modular architecture, which supports multiple child chains and offers a lightweight API for integration (Adarve et al., 2024). The framework ensures scalability and security while providing an accessible interface for the implementation of the blockchain-based archiving system. The underlying consensus mechanism, proof-of-stake (Saleh et al., 2024), enables efficient and energy-conscious transaction validation. The mechanism ensures the integrity and immutability of the archived data while maintaining compatibility with the lightweight nature of SHM systems.

Table 1: Enumeration data types

Enumeration	Description and value range
SensorNodeDataTypeEnum	Characterization of an instance of the class *SensorNodeData* **Value range:** *Component-removal, component-mounting, component-check, sensor-node-event*
EventTypeEnum	Subcharacterization of an instance of the class *SensorEvent* (Attribute type: sensor-node-event) **Value range:** *Heartbeat, alert, measurements*
AlertTypeEnum	Characterization of an alert state in a *SensorNode*. **Value range:** *Bias, drift, gain, precision-degradation, complete-failure*

The system architecture of the blockchain-based archiving system is shown in Figure 2. Blockchain nodes, built on the *Ardor* framework, are deployed on a central server. The blockchain nodes themselves can be accessed by other components via the *Ardor API*; other components include the data-generating components described in the use cases, such as the sensor nodes (related to faulty data,

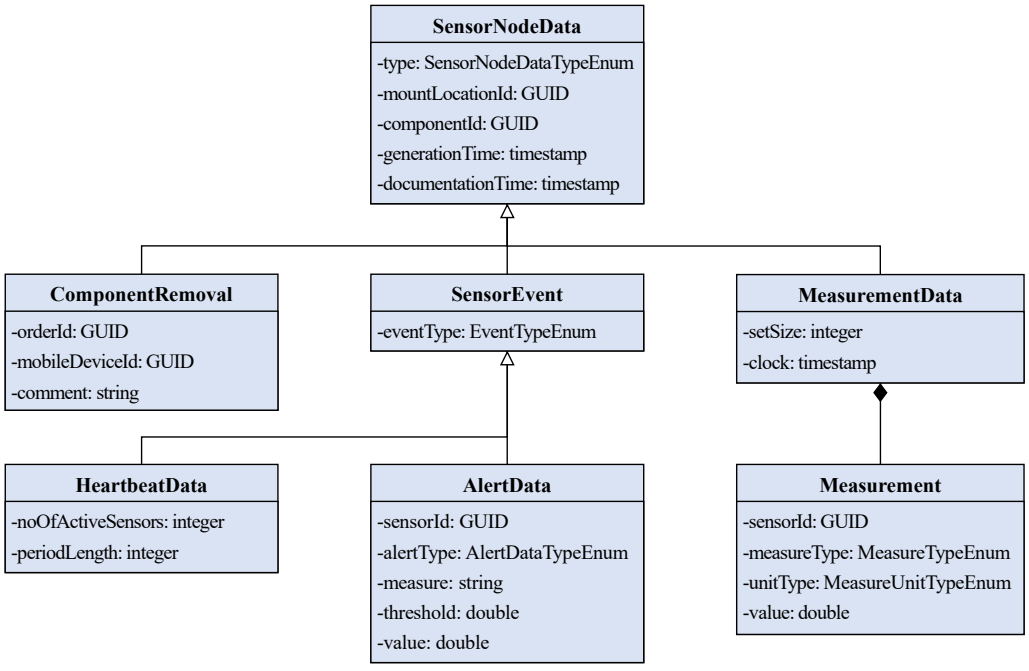

Figure 1: Data model describing the sensor node data stored in a blockchain (excerpt)

use case 1) and mobile devices (related to sensor replacements, use case 2), as well as evaluating controlling components, such as personal computers, that also access the blockchain nodes. Clients run on mobile devices that document the changes made to the sensor nodes. Clients use a *Semantic API* for documenting changes, which converts the client calls into calls to the *Ardor* blockchain framework. The sensor nodes themselves also host client software that transmits the data to be documented to the *Ardor API* on the blockchain nodes. On personal computers, a control client is utilized to allow querying and evaluating the data documented in the blockchain. The blockchain node is connected to the web via the *Ardor* blockchain framework and interacts with further *Ardor* blockchain nodes on the web.

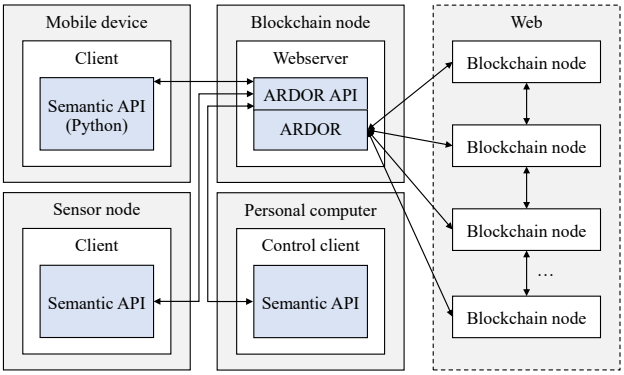

Figure 2: Architecture of the blockchain-based archiving system

At the core of the implementation lies the *Semantic API* software component written in Python. The *Semantic API* translates the sensor node data derived from the use cases (i.e. use case scripts based on Figure 1) into specific calls to the *Ardor API*. The component acts as a middleware, ensuring smooth communication between the data-generating components (e.g., sensor nodes and mobile devices) and the blockchain nodes operating on the *Ardor* framework (Figure 3). The *Semantic API* enables automatic conversion of semantic descriptions into blockchain transactions, ensuring that all stored data adheres to the tamper-proof and audit-compliant requirements of the SHM system. Table 2 summarizes the functions of the elements of the blockchain-based archiving system, as shown in Figure 3. In the following section, the validation of the blockchain-based archiving system is presented.

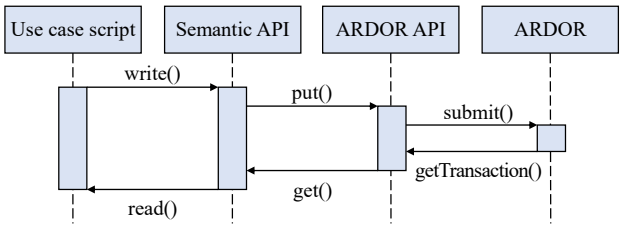

Figure 3: Sequence diagram of calls to the Ardor blockchain framework

Table 2: Elements of the blockchain-based archiving system

Element	Function
Use Case Script	The script is written in Python and functions as an interface for user request (i.e. write and read calls).
Semantic API	The *Semantic API* is written in Python as an API wrapper for tagged data operations in the *Ardor API*. Tagged data operations allow users to upload and retrieve data to the blockchain. The *Semantic API* maps the use case script requests to the *Ardor API* as put and get calls.
ARDOR API	The *Ardor API* is provided by the blockchain nodes and communicates with the blockchain via HTTP requests (i.e. submit and get transaction calls). The *Ardor API* is installed on a central server in a secure local network and receives requests from clients via the *Sematic API*.
ARDOR	*Ardor* network with all blockchain nodes.

Validation of the blockchain-based archiving system

In this section, the blockchain-based archiving system is validated to test functionality and performance with a proof of concept. The functionality of the archiving system is assessed by observing storage and querying functionalities, while the performance is assessed by observing the latency and the cost of transactions for storage and querying.

An SHM system comprising four sensor nodes, labeled WSN1, WSN2, WSN3 and WSN4, is used for the validation tests (Figure 4). The SHM system is installed on a test structure and monitors the acceleration of the structure caused by a shake table. Faulty data, representing a “bias” sensor fault, is induced in sensor node WSN1 to trigger an alert and the faulty sensor is then replaced for a new one. As described in Figure 2, each sensor node in the SHM system, the mobile device, and the personal computer use the *Semantic API* as middleware to communicate with the blockchain nodes via the *Ardor API* hosted in a webserver. Data is exchanged using a JSON data format. The use case scripts shown in Figure 3 are used to write and query the sensor node data into the blockchain, e.g. (i) the measurement data and the alert associated to the faulty data for use case 1 as well as (ii) the sensor replacement on the sensor node affected by the faulty data for use case 2.

For the proof of concept, transactions are transmitted to the blockchain documenting any alerts that have been triggered due faulty data during a period of 1 hour, for a total of two alerts. When the “bias” sensor faults are detected in WSN1, the corresponding measurement data is stored alongside the alert with a maximum transaction

size of 42 KB. The sensor replacement for WSN1 is documented once using the mobile device with a maximum transaction size of 1 KB. Once all storing transactions have been written to the blockchain, a query transaction is preformed to read the data stored in the blockchain to evaluate the functionality of the archiving system. A maximum latency threshold per transactions is defined as 10 min.

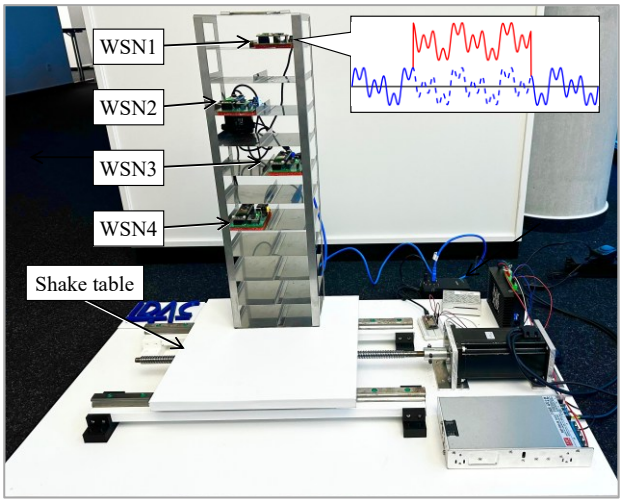

Figure 4: Validation test for an SHM system presenting a bias sensor fault in one sensor node

The data submitted as transactions to the blockchain is visualized via the interface of the *Ardor API* of a local *Ardor* blockchain node (Figure 5). The interface also documents recent transactions triggered by the use case scripts and confirmed by the other nodes of the blockchain. The sensor node data stored in the blockchain is retrieve using the query methods in the use case scripts and visualized on the *Ardor API*. Each transaction has an average cost of 0.23 Ignis.

Latency values apply to writing to and reading from the blockchain for storing and querying data. Writing to the Ardor blockchain in storing transactions comprises two steps: i) *submitting transactions* to a child chain with the data to be stored and ii) broadcasting the transactions to the network of blockchain nodes for *block confirmation*. When a storing transaction is submitted, nodes of the child chain validate the transaction; e.g., the nodes check whether the storing transaction has been authenticated by cryptographic signatures. If the validation is positive, the storing transaction is bundled with other transactions awaiting inclusion in a block. A block is forged on the parent chain using the proof-of-stake consensus to add the bundled transactions from the child chain into the block. Then, the block is added to the parent chain, securing the storing transaction, and the block is broadcasted to the network. When a certain number of new blocks (here 10 blocks) have been added to the parent chain, the storing transaction is considered to be confirmed. Reading from the Ardor blockchain in querying transactions comprises the *retrieval of historical transactions* by specifying the unique identifier of a transaction.

Table 3 shows the mean latency values of three measurements, transaction submission, block confirmation, and retrieval of historical transactions. It should be noted that retrieving historical transactions applies to a node for which the information has not yet been pruned from the blockchain. Blockchain pruning is a technique used to help maintain efficiency by pruning child chain transactions. If the retrieval time exceeds a predefine threshold, the data can only be found in archival nodes.

Table 3: Mean latency values for storing and querying transactions in the blockchain

Latency	Local blockchain node
Transaction submission	0.0145 s
Block confirmation (10 blocks)	186.1613 s
Retrieval of historical transactions	0.0152 s

Writing operations in the form of transaction submissions are observed to be efficient, while the block confirmation takes several minutes. Until the transaction is confirmed, there is a potential risk of blocks being discarded, making a new transaction submission necessary. Discarded blocks have not been observed in this study; however, software logic must take discarded blocks into account.

The latency values show satisfactory performance, in which local blockchain nodes present low latencies. Local blockchain nodes present minimal overhead in submitting transactions when publishing requests and may access the blockchain database efficiently when retrieving historical transactions. The confirmation time, in which data is replicated to other blockchain nodes, may vary between transactions due to fluctuations in block creation intervals. Furthermore, the following aspects are noted:

- The sensor node data relevant to faulty data and sensor replacements could be stored and queried successfully in the *Ardor* blockchain. Storing sensor node data for sensor malfunctions is considered a lightweight transaction as the transactions include limited data (Figure 1) and are low frequency events.
- Local blockchain nodes can be used for writing and reading. Via the interface of a local *Ardor* blockchain node, it is observed that the sensor node data is also included in the blockchain of subsequent blockchain nodes. A remote node can be used for writing on site owing to the online capabilities of the Ardor framework.
- The online functionalities of the *Ardor* framework facilitate deploying data-generating components for on-site documentation and controlling components for queries.

- The edge-computing capabilities of the sensor nodes support the execution of fault diagnosis algorithms on-board and of the *Semantic API* for writing sensor data to the blockchain. Similarly, mobile devices and personal computers can send requests using an HTTP protocol to the *Ardor API* via the *Semantic API*.
- The latency to querying data from the blockchain is suitable for retrieving sensor node data in a blockchain. However, the latency for storing data to the blockchain showcases that the software logic should be able to handle discarded blocks by repeating the transaction submission.
- Transaction costs are affordable. Each transaction for writing and querying requires a transaction fee, which is usually paid by the user who initiated the transaction. For the transactions involved in both use cases, a fee of 0.23 Ignis per transaction could be successfully completed by the user initiating the transactions. During the validation test, the exchange rate for 1 Ignis was €0.0009134 (i.e., 4760 transactions for 1€). A tradeoff may be considered between transaction costs and frequency in SHM applications.

Consequently, the proposed blockchain-based archiving system may facilitate the integration of IoT frameworks and cloud services frequently used in SHM applications. Furthermore, the proof of concept showcased that lightweight transactions show suitable performance for SHM applications. However, drawbacks of the blockchain-based archiving system have been observed that may limit the integration with BIM tools and digital twins. The main drawbacks are described as follows:

- The method for storing transactions in the archiving system is time-consuming and shows latency values that might cause synchronization issues in case BIM tools or digital twins are connected to the system. Hence, the Ardor framework is not suitable for real-time data exchanges hindering feedback loops frequently used in digital twinning.
- Since the *Ardor* framework is a public blockchain framework, the data is publicly available, requiring encryption methods that delay the translation calls between the SHM system and the blockchain-based archiving system via the *Semantic API*.
- Transaction pruning in the *Ardor* framework may affect long-term data storage, requiring additional applications to ensure the availability of blockchain history.
- The parent chain in the *Ardor* framework handles consensus for all child chains, which may cause scalability bottlenecks with increasing number of transactions. The scalability constraints limit the application of the *Ardor* framework for on-chain storage of high-volume data, such as sensor data with a high-frequency rate, or large-size files, such as BIM files.

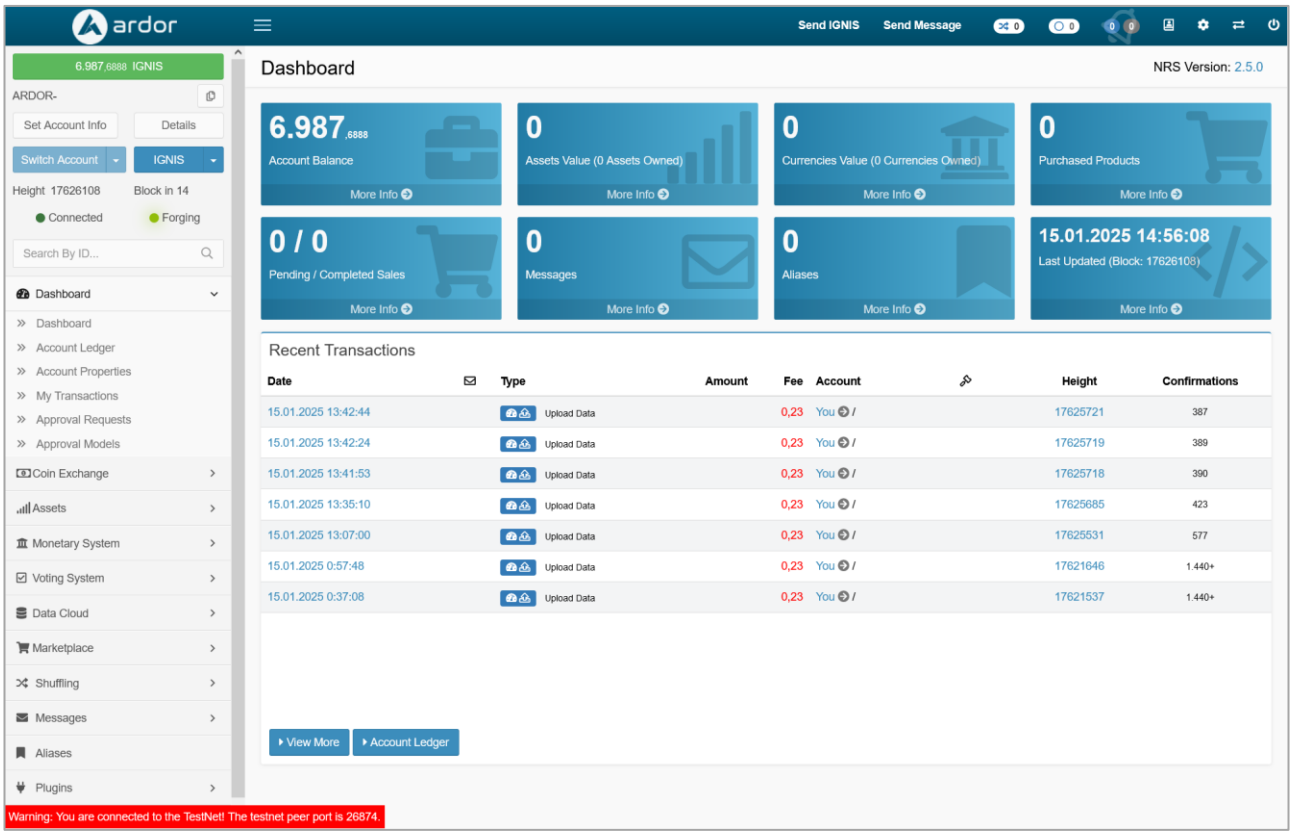

Figure 5: Screenshot of the interface of the local Ardor blockchain node

Compared to other archiving technologies, the *Ardor* framework used in the archiving system presents a suitable solution that can be easily deployed and integrated to SHM systems and cloud services as modular add-on for archiving of sensor malfunctions. However, transaction delays and block size limit the applicability of the *Ardor* framework for real-time data exchanges and high-volume data management in SHM applications. When integrating BIM tools and digital twins, an InterPlanetary File System protocol (Nyaletey et al., 2019) may be deployed for supporting off-chain distributed data sharing of large-size files with data traceability. Other distributed ledger technologies, such as IOTA (Shabandri et al., 2019) and Hedera Hashgraph (Krasnoselskii et al., 2020), provide a better solution compared to the *Ardor* framework for real-time data exchanges with parallel transactions and may be explored to provide a more flexible archiving system for SHM applications.

Summary and conclusions

This paper has presented a blockchain-based archiving system for sensor malfunctions, designed as a modular add-on for SHM systems. Using a public blockchain framework, the blockchain-based archiving system has been implemented and validated for two use cases common in SHM, (i) faulty data and (ii) sensors replacements via a proof of concept. Specifically, an SHM system comprising four sensor nodes has been used to test the blockchain-based archiving system in terms of functionality and performance by inducing faulty data in one of the sensor nodes and by documenting a sensor replacement. Validation tests have been conducted with respect to functionality and performance of the system. The functionality has been validated by observing the translation of the sensor node data to the blockchain framework via the *Semantic API*, which facilitated documenting and querying (i) the measurement data and the alert associated to the faulty data as well as (ii) the sensor replacement on the sensor node affected by the faulty data in the blockchain. The performance has been assessed by observing the latency of the transaction in the blockchain, where storing transactions have shown to be more time-consuming than querying transactions.

In conclusion, by employing the *Ardor* framework, the blockchain-based archiving system with online functionalities has been developed for SHM use cases frequently encounter in the real world. The integration of SHM systems with edge-computing capabilities and blockchain-based data management systems has been addressed by mapping client requests to operations in the Ardor framework via the *Semantic API*. The validation results have demonstrated the potential of blockchain technology to securely archive sensor malfunctions in SHM systems online. The results have highlighted the functionality and performance of the blockchain-based archiving system in addressing the challenges of securely archiving sensor malfunctions in SHM systems. The *Ardor* framework has facilitated developing an application that seamlessly integrates with SHM systems for exchanging data in a secure, and decentralized manner, though drawbacks may limit scalability. By leveraging blockchain technology, the archiving system has provided tamper-proof and verifiable data storage, enabling secure tracking of sensor malfunctions, while maintaining audit-compliant documentation. The modular design of the archiving system has ensured compatibility with existing SHM systems, demonstrating its practical applicability. Moreover, the validation has confirmed the suitability of the blockchain-based approach for supporting documentation of both faulty data and sensor replacement in SHM systems.

Future work may be directed towards enhancing the blockchain-based archiving system to accommodate large-scale SHM deployments involving a significantly higher volume of sensor data inspired by tangle ledger technology. In addition, integrating predictive analytics with the blockchain-archived data may be explored to enable proactive maintenance strategies, further increasing the functionality and performance of the blockchain-based archiving system. Last, but not least, further validation in real-world SHM applications, encompassing diverse environmental and operational conditions, may be pursued to ensure the robustness and adaptability of the proposed system across various scenarios.

Acknowledgements

The authors gratefully acknowledge the financial support provided by the German Research Foundation (DFG) through grant SM 281/22-1 and the German Federal Ministry for Digital and Transport (BMDV) within the mFUND program under grants 01FV2013B and 01FV2059C. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of sponsors mentioned above.

References

Al-Nasser, H., Al-Zuriqat, T., Dragos, K., Chillón Geck, C., & Smarsly, K. (2024). Identification of combined sensor faults in structural health monitoring systems. Smart Materials and Structures, 33(8), 085026.

Al-Zuriqat, T., Chillón Geck, C., Dragos, K., & Smarsly, K. (2023). Adaptive fault diagnosis for simultaneous sensor faults in structural health monitoring systems. Infrastructures, 8(3), 39.

Adarve, O.A., Bianzino, A.P., Frendanez Muñoz, P., Abdallah, R., Garcia Garcia, R., Fernandez, R., & Garcia Recuero, A. (2024). How to choose a blockchain technology for an innovation project: Taxonomy and use cases. In: Proceedings of the 9th Future Technologies Conference, London, United Kingdom, 11/14/2024.

Bartels, J.-H., Potthast, T., Möller, S., Grießmann, T., Rolfes, R., Beer, M., & Marx, S. (2024). Robust SHM: Redundancy approach with different sensor integration levels for long life monitoring systems. In: Proceedings of the 11th European Workshop on Structural Health Monitoring, Postdam, Germany, 06/10/2024.

Brötzmann, J., Panda, J., & Rüppel, U. (2022). Blockchain Technology as a Monitoring Tool for Sensor Data. In: Proceedings of the 19th International Conference on Computing in Civil and Building Engineering. Cape Town, South Africa, 10/26/2022.

Chen, J., Reitz, J., Richstein, R., Schröder, K.-U., & Roßmann, J. (2024). IoT-based SHM using digital twins for interoperable and scalable decentralized smart sensing systems. Information, 15(3). 121.

Crosby, M., Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). Blockchain technology: Beyond bitcoin. Applied Innovation, 2(6–10), 71.

Dragos, K. & Smarsly, K. (2016). Distributed adaptive diagnosis of sensor faults using structural response data. Smart Materials and Structures, 25(10), 105019.

Deng, Y., Zhao, Y., Ju, H., Yi, T.H., & Li, A. (2024). Abnormal data detection for structural health monitoring: State-of-the-art review. Developments in the Built Environment, 17(2024), 100337.

Ding, Y., Han, R., Liu, H., Li, S., Zhao, X., & Yu, Y. (2016). Bridge inspection and management system based on smartphone. In: Proceedings of the ASME 2016 Conference on Smart Materials, Adaptive Structures and Intelligent Systems, Stowe, VT, USA, 09/28/2016.

Fritz, H., Peralta, J., Legatiuk, D., Steiner, M., Dragos, K., & Smarsly, K. (2022). Fault diagnosis in structural health monitoring systems using signal processing and machine learning techniques. In: Cury, A., Ribeiro, D., Ubertini, F., Todd, M. D. (eds.). Structural health monitoring based on data science techniques. Pp. 143-164. Cham, Switzerland: Springer.

Gigli, L., Sciullo, L., Montori, F., Marzani, A. & Di Felice, M. (2022). Blockchain and Web of Things for structural health monitoring applications: A proof of concept. In: Proceedings of the 2022 IEEE 19th Annual Consummer Communication & Networking Conference, Las Vegas, NV, USA, 01/08/2022.

Jelurida Swiss SA. (2024). Ardor. Accessed 06/14/2024 from https://www.jelurida.com/ardor.

Krasnoselskii, M., Melnikov, G., & Yanovich, Y. (2020). Distributed Random Number Generator on Hedera Hashgraph. In: Proceedings of the 2020 3rd International Conference on Blockchain Technology and Applications, 12/14/2020, Xi'an, China.

Lai, Y., Tong, L., Liu, J., Wang, Y., Tang, T., Zhao, Z., & Qin, H. (2022). Identifying malicious nodes in wireless sensor networks based on correlation detection. Computers & Security, 113(2022), 102540.

Law, K.H., Smarsly, K., & Wang, Y. (2014). Sensor data management technologies for infrastructure asset management. In: Wang, M.L., Lynch, J.P., Sohn, H. (Eds.). Sensor Technologies for Civil Infrastructures (1st edition). Pp. 3-32. Sawston, UK: Woodhead Publishing.

Moridi, E., Haghparast, M., Hosseinzadeh, M., & Jafarali Jassbi, S. (2020). Fault management frameworks in wireless sensor networks: A survey. Computer Communications, 155(2020), pp. 205-226.

Nyaletey, E., Parizi, R.M., Zhang, Q., & Choo, K.-K.R. BlockIPFS - Blockchain-Enabled Interplanetary File System for Forensic and Trusted Data Traceability. In: Proceedings of the 2019 IEEE International Conference on Blockchain, Atlanta, GA, USA, 07/14/2019.

Ramasamy, L.K., Khan, F., Imoize, A. L., Ogbebor, J.O., Kadry, S., & Rho, S. (2021). Blockchain-based wireless sensor networks for malicious node detection: A survey. IEEE Access, 9(2021), pp. 128765-128785.

Saleh, F. (2021). Blockchain without Waste: Proof-of-Stake. The Review of Financial Studies, 34(3), pp. 1156-1190.

Shabandri, B. & Maheshwari, P. (2019). Enhancing IoT security and privacy using distributed ledgers with IOTA and the Tangle. In: Proceedings of the 6th International Conference on Signal Processing and Integrated Networks, Noida, India, 03/07/2019.

Smarsly, K. & Law, K. H. (2014). Decentralized Fault Detection and Isolation in Wireless Structural Health Monitoring Systems using Analytical Redundancy. Advances in Engineering Software, 73(2014), pp. 1-10.

Steiner, M., Legatiuk, D., & Smarsly, K., (2019). A support vector regression-based approach towards decentralized fault diagnosis in wireless structural health monitoring systems. In: Proceedings of the 12th International Workshop on Structural Health Monitoring. Stanford, CA, USA, 09/10/2019.

Xie, X., Wang, J., Han, Y., & Li, W. (2024). Knowledge graph-based in-context learning for advanced fault diagnosis in sensor networks. Sensors, 24(24), 8086.

Yu, X., Fu, Y., Li, J., Mao, J., Hoang, T., & Wang, H. (2024). Recent advances in wireless sensor networks for structural health monitoring of civil infrastructure. Journal of Infrastructure Intelligence and Resilience, 3(1), 100066.

Zhang, D.-M., Nie, C., Zhang, J.-Z., Huang, H.-W, & Huang, X. (2024). Consortium blockchain-based tunnel data bank for traceable sharing and treatment of structural health monitoring data. Automation in Construction, 167(2024), 105720.