Giao *et al. Journal of Cloud Computing* (2026) 15:34
https://doi.org/10.1186/s13677-026-00842-4

Journal of Cloud Computing

RESEARCH Open Access

Enhancing health data integrity using Distributed Ledger Technology

Joao Giao[1*], Fernando Luis-Ferreira[1], Joao Sarraipa[1] and Ricardo Jardim-Gonçalves[1]

Abstract
Nowadays, with the increased integration of cloud-computing, data integrity continues to be a problem in the current eHealth sector. This security principle is considered fundamental to ensure the accuracy and reliability of data by ensuring protection from unauthorized and illicit tampering. The present work aims to demonstrate the ability for the Distributed Ledger Technology (DLT) to provide trust and confidence in the healthcare infrastructure for patients, healthcare professionals and policy makers. The DLT has the potential to become one of the most reliable solutions for the many challenges facing the healthcare industry for its potential to enable more secure, transparent, and equitable data management. Although much documentation exists about applications in this domain, it is mostly presented in high-level conceptualization, without detailing the actual development or implementation. This document proposes a metadata-based approach to protect healthcare data integrity in compliance with GDPR, ensuring trustworthy data access for end-users, while demonstrating that the solution can be deployed on low-resource hardware with minimal adaptation effort and time constraints.

Keywords Blockchain, Data integrity, Distributed Ledger Technology (DLT), Raspberry Pi, Smart contracts

Introduction

With the evolution of technology, healthcare has become more widely accessible throughout the world and both its efficiency and safety have drastically improved. Large amounts of health data are being collected and made available through existing and emerging technological tools and media, to allow patients to possess the control of generating, managing, and sharing Electronic Health Records (EHRs) with family, healthcare providers and other authorized data consumers. Generally, EHRs are medical and clinical structured data related to a given patient and stored by a reliable healthcare provider [1], as a means to improve patient care and their well-being.

*Correspondence:
Joao Giao
jgs@uninova.pt
[1]NOVA School of Science and Technology, Center of Technology and Systems (UNINOVA-CTS) and Associated Lab of Intelligent Systems (LASI), NOVA University Lisbon, Street, Caparica, Lisbon 2829-516, Portugal

However, the risk of data compromise or data loss exponentially increases as organizations rely on these digital technological advances [2]. Several regulations, including the General Data Protection Regulation (GDPR), have been implemented to improve security and transparency in storing and processing such information. These regulations aim to address the growing need for openness regarding access to personal health data, thus empowering individuals as the owners of their own data. This is an important factor in today's health systems, as it provides patients with the choice of giving consent to trusted doctors or healthcare organizations.

Patients' data are constantly being transmitted across different areas during life events, causing EHRs to move from one service provider database to another. Data are stored, processed, and manipulated with third parties to provide services for the system's users, thus, being exposed to tampering. Therefore, data integrity is an important security concern as it refers to the

completeness and exclusion of unintended modifications during the system's data lifecycle. Ultimately, it ensures that data are provided accurately to the systems components and end-users. Unintended modifications could arise, for instance, due to malicious or unintentional human behaviour, or from technical imperfection, damage, or loss of a technical infrastructure. Consequently, preventing data from security breaches and modification attacks is a challenging task [3], especially for health professionals but also researchers and developers, as new threats are always on the rise. Attackers specifically target healthcare sub-domains to manipulate valuable data or use the breached data to orchestrate other attacks on systems [4], creating larger ramifications of data integrity breaches that are often unknown to the data owner or hospitals and medical institutions. The consequences of tampering healthcare data and information can be dangerous and include rejection of all data from a trial, incorrect conclusions, diminished credibility for the health institution and can cause a life-threatening situation for the patient's data. As a result, it is necessary to guarantee the health data used by the system remains in its original form.

As previously stated, EHRs are particularly vulnerable and traditional integrity verification methods, mainly based on hash functions or asymmetric cryptographic algorithms, often rely on Third Party Auditors (TPAs) to execute auditing tasks. The majority of healthcare systems use centralized client-server based architectures, where a central authority has full-access to the system, which requires centralized trust [5]. However, not only the system completely depends on the credibility and trustability of the TPA, but also increases the possibility of user's private information leakage as well as creating a single point of failure. With a decentralized and consensus-driven nature, Distributed Ledger Technology (DLT) has recently emerged as one of the most promising technologies garnering significant interest for its transparency, immutability, security and decentralization [6]. Typically, a DLT shares all data in the form of transactions, and once a consensus is reached among the participants to approve one transaction, it becomes available for all nodes that are participating in the network [7]. DLTs can also support smart contracts, which allows to automatically run pre-defined scripts. Hence, due to the nature of the network, cryptographic techniques, time-stamped records, and its data transparency, this recent technology ensures data are immutable and auditable.

The process of data collection and handling of sensitive data requires special procedures and strategies to ensure its security is accomplished. Considering the security issues associated with complex systems such as in the healthcare domain, the primary goal of our study was to ensure the data used by an health system remains unaltered, i.e., the integrity of the data is preserved during its lifecycle. Therefore, this work aims to answer to the following research question: How can a health information system support controlled data sharing and protect data integrity using a metadata-based approach, while complying with GDPR and operating efficiently on resource-constrained devices? The following main contributions have been made in this paper:

- a two-layer DLT-based architecture for healthcare data integrity, composed of a healthcare layer which includes various components and mechanisms used for data security and data privacy functionalities and a DLT layer that provides decentralized trust, tamper-evidence, and auditability by maintaining shared and consistently validated information across all participating nodes. This architecture enables secure, trustworthy, and GDPR-compliant management of electronic health data;
- a metadata model for health data that supports secure and privacy-preserving integrity validation. The purpose of this metadata model is to enable the management of sensitive data without exposing personal information, while still providing sufficient information to verify authenticity, detect tampering, and ensure trustworthy use of the underlying health resources;
- full prototype implementation and integrated it into a real-world clinical use case from the FAITH (a Federated Artificial Intelligence solution for moniToring mental Health status after cancer treatment) project [8], complemented by a comprehensive performance evaluation. This prototype demonstrates the feasibility, operational compatibility, and effectiveness of the proposed solution under realistic healthcare conditions.

Background

Data acquisition and its management are at the centre of both healthcare institutions and users, the citizens. Data, properly acquired and managed, provide the ground for the healthcare institutions to develop the best assessment of a citizen's health, deploy the proper assets and to establish the best treatment plan for each person's specific needs. To achieve such an endeavor, the DLT can be a tool to provide such guarantees and to promote trust and reliability of healthcare data management systems. Distributed ledgers are a form of technology that uses a public or private network to distribute, exchange, and store data among users. At its core, it enables an inviolable and incorruptible means to securely store relevant information.

Security and privacy of health systems

Security and privacy in health systems play a central role in safeguarding the integrity of healthcare data, preserving the confidentiality and trust, essential to deliver optimal patient care.

Medical devices and health services can provide opportunities to monitor ongoing diseases, share clinical information and treatments with patients or caregivers, and give feedback to the care team. This data sharing allows to have an overall understanding of existing clinical modalities and the situation of the global healthcare system, allowing to take actions accordingly to the present context. In an hospital environment, early commencement of treatment and medication, and the decisions on how to proceed, depend on patients' data in the different modalities available. In pursuing such a goal, it is necessary to use security and privacy practices to maintain the patient trust and reinforce health system's reluctance to share data.

Many countries are enhancing their healthcare services through the application of information technology. Recent developments in this area have led to the digitalization of health records, thereby creating new and improved ways to efficiently collect, process, store, consult, and share health information [9]. This tendency is driven by the fact the digitalized health information is more portable and can be easily shared among healthcare organizations. Surveys and administrative registers form a basis for the national health monitoring and for this reason, they are considered as an essential component of surveillance by providing a diverse and detailed picture of a patient/population health status and trends over time [10].

In principle, the patient, source of health data, controls access to and use of the data extracted from their healthcare and interaction with digital platforms. In practice, the health institution is the organization entity that collects and manages the patient data, therefore, the protection of data and user's privacy is important and must be ensured by the health system. A patient needs to trust that health data referring to him is properly managed, ensuring privacy and data security. In this process, there is legislation to be followed and regulations to be observed in the interaction between the citizen and the health entity, namely GDPR in Europe. This regulation was designed to harmonize and define data privacy laws across Europe (and the companies doing business in Europe) to protect and empower EU citizens' personal data and reshape the way organizations handle data. In particular, GDPR sets mandatory requirements health systems as well the health data exchange, namely [11]: data protection by design and by default - only personal data which is necessary for each specific purpose of the processing, are used by the system; data portability - data owner has the right to take their data to other system, and for this reason, the data controller must provide its data in a machine-readable form; right to be forgotten - data owner has the right to obtain from the data controller the immediate erasure of their data; unambiguous consent - data owner agreement for processing and storing their data. It must be freely given, specific, informed and unambiguous. The data controller must also provide information of how the information will be processed; right to access/records of processing activities - requirement for maintaining data provenance, i.e., the processes and methodologies the data were subjected to.

Health systems involve collecting, storing and analyzing data related to patient's health, diagnosis, and treatment. In this case, the authenticity of health information is essential to maintain the trust of patients and healthcare institutions. Thus, data integrity is an important factor in almost any data and computation context as well as an important part of data security and privacy [2]. Protecting the data integrity of a system involves safeguarding it against unauthorized deletion or modifications, either by unauthorized users or unauthorized software, through the data lifecycle. Thus, data integrity is crucial to ensure the accuracy, completeness, and consistency of data [12]. Moreover, the system should be able to detect and recover improper alterations of the data [13]. To that purpose, it is necessary to analyse the system and identify the possible vulnerabilities of the data lifecycle. However, the healthcare industry is still lagging behind from other industries in its efforts to protect the data integrity of its stakeholders, since current medical records are stored in centralized or federated databases, maintained by hospitals, insurers, or governmental agencies [14]. Cryptographic mechanisms, like symmetric or asymmetric key based approaches or hash functions, enhance integrity assurances for medical records [15]. Another security tool a system can use to detect data integrity is the checksum algorithm, however, those only provide protection against accidental errors, such as errors that can occur through the noise in a communication channel [16].

What is a DLT?

A distributed system is composed by a collection of nodes that can communicate through a network and be able to behave independently of each other. A node mentions a computing element, which can be either a hardware device or a software process. Although different nodes belong to the same system, their users (people or applications) see all nodes as a single logical system. This behaviour is possible because autonomous nodes communicate through a network and collaborate with each other in a coordinated fashion (e.g., to cooperatively run a distributed application) [13]. Each node connects using a unique address and use the gossip protocol to exchange

network information, such as blocks, transactions, and addresses [17]. A ledger is used to mention a group of electronic records, held by a significant proportion of the network participants, that were previously approved as true and are unlikely to be erased or amended during its lifecycle. The technology that enables the operation and use of distributed ledgers is called DLT. Usually, this information is owned by every node of a P2P network, where their users have available a consensus mechanism that guarantees the agreement of what is stored between the distributed nodes. Conceptually, DLT is secure by design and offers the capability to achieve decentralized consensus and consistency, while also demonstrating resilience against both intentional and unintentional attacks. The most useful properties [18, 19] a DLT can provide for an healthcare system are: shared record-keeping - multiple entities are allowed to collectively create, maintain, update and made accessible for those who need a shared set of records (ledger); multi-party consensus: a group of parties of the DLT contribute to reach an agreement on the records to be stored and shared on the ledger; independent validation - each participant is allowed to independently verify their transactions as well as the integrity of the stored information; tamper evidence: data should not be modified, and each participant can perceive if non-consensual changes occurred; tamper resistance: as a distributed system, the higher number of the nodes, the most difficult it is to change past records, increasing the integrity of the stored data on the ledger; anonymity and programmability - the identity of users is unidentifiable, allowing to safeguard their privacy. The programmability feature enables automation of new transactions through smart contracts, based on data from the ledger [20].

DLT in healthcare

Nowadays, security and privacy are two of the primary concerns of the healthcare industry, due to the tremendous amount of health data accessed and transmitted over the internet [3]. In this domain, security is used to protect system's data in transit and at rest, to serve health monitoring purposes. Traditionally, health data are stored in physical form and any data tampering can be detected. In the case of digital data, the most difficult challenge arises from its susceptibility to alteration or deletion. Conventionally, current solutions may be enough, however, this sector requires more transparency, correctness, validity, authenticity, and reliability [21]. They are complex systems which makes the adaptation of new technologies more difficult. Most of the existing healthcare systems are subjected to centralized computation, storage, and processing. Such centralization can be problematic, as facilitates a single point of failures and information leakage due to the increase of cybersecurity attacks [22]. Other solutions rely on external and independent entities to publicly verify the integrity of data, such as a TPA or use a database which can provide both privacy and tamper-resistant features as discussed in [23]. However, such solutions must assume the TPA behaves in an honest way, which is not a correct assumption [24]. As an example, it may be tempting for hospitals or insurance companies to hide or replace health records. For these reasons, DLT has gained great consideration and application in recent years. As such, numerous studies have been published that applied DLTs in the domain of this work. For example, in [25], the authors describe an access management system for personal health records, using the IOTA DAG framework. Focused on personal health records and data from IoT devices, four different applications were created that interact with the system's DLT. They focused on hospital admission, patient discharge, and remote patient health data records when registering and retrieving personal data. The IOTA is used to store the external database hashes location, interact with smart contracts, and communicate with IoT devices. In [26], the authors developed a platform which captured data streams from IoT health wearable devices and stored them on a DLT. Separated by different modules, their platform allows the pairing of wearable devices, capturing and verifying their data streams, and publishing them on a dedicated DLT infrastructure. The DLT based integrated healthcare solution for Bangladesh proposed in [21] is a blockchain-based solution that uses a private Hyperledger Fabric framework. There system uses Bangladesh's public and private healthcare service providers and is focused on handling secure digital agreements for commerce or collaboration, by utilizing the data immutability and smart contracts features. A proof of concept was presented which included handling licensed medications, administrative services, official health data, patient medical histories, and licensed pharmaceutical data, which were stored in the system. There was a concern about GDPR and privacy, e.g., the private data were stored in a private database on the peers of authorized organizations, which was accessed only by authorized peers. The authors in [27] present an architecture that uses a blockchain network to facilitate communication between healthcare providers by validating the shared data. They also analyse the monetary costs associated with various public blockchain services available for the system. In Namasudra2023, the authors propose an architecture for managing medical certificates, where the data is stored off-chain and the DLT is used to ensure its integrity. This work primarily focuses on analyzing the gas costs and their monetary implications for certificate management, without addressing privacy regulations such as HIPAA or GDPR. Table 1 summarizes

Table 1 Comparative analysis of existing DLT solutions in the healthcare domain

Reference	DLT type	Type	Smart contract	GDPR	Authorization	Backup	Maturity level
[25]	IOTA	Private	Yes	No	Yes	No	Proof of concept
[26]	IOTA	Public	No	No	No	No	Prototype
[21]	Hyperledger Fabric	Private	Yes	Yes	Yes	No	Proof of concept
[27]	Ethereum	Private	Yes	No	Yes	No	Proof of concept
[28]	Ethereum	Public	Yes	No	Yes	No	Proof of concept
Proposed solution	Ethereum	Private	Yes	Yes	Yes	Yes	Prototype

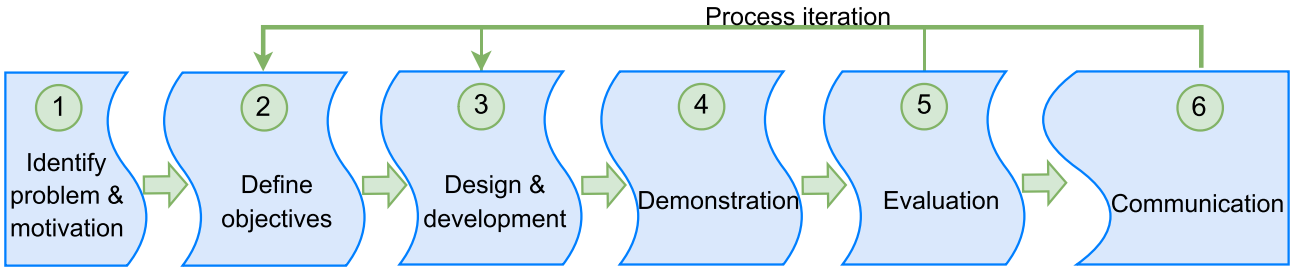

Fig. 1 DSR methodology adopted to develop this study architecture, based on [29]

existing healthcare system solutions that integrate DLT functionalities.

In general, there is limited emphasis on data privacy and users' data rights, and many of the reviewed solutions do not take existing legislation—such as GDPR or HIPAA—into account. Furthermore, the software development of these solutions is often in early stages, with the majority of the works being theoretical or limited to architectural proposals. Finally, most of the solutions listed in Table 1 lack sufficient implementation details, making their practical adoption or replication difficult.

Methodology

This section presents the research methodology adopted to design the architecture for protecting health data integrity in a health information system. This study follows a Design Science Research (DSR) methodology [29] which consists of several iterative steps, as illustrated in Fig. 1.

- problem identification and motivation - The first step in this methodology involves defining the specific research problem and justify the value of a solution. In this work, the identified issue is the absence of a trustworthy and tamper-evident mechanisms to guarantee the integrity of electronic health data throughout its lifecycle. Sections "Introduction" and "Background" of the manuscript describe these challenges and motivate the need for a new approach to assure health data integrity;
- define the objective(s) for a solution - From the problem definition, the objective(s) that a successful solution must fulfill are inferred in Section "Background". The global objective of this research is to design and validate a DLT-based mechanism that enables secure, verifiable, and privacy-preserving data-integrity validation, ensuring GDPR-compliant processing, and maintaining operational compatibility with healthcare environments;
- design and development - In this step, an artifact that addresses the defined objectives is created. In this work, the artifact consists of the proposed DLT-based architecture and its associated components. Section "Proposed solution - architecture design" details the design of:
  - an architecture that coordinates the necessary components and mechanisms to assure the integrity of health data;
  - a metadata model that supports privacy-preserving management of sensitive data to comply with GDPR regulations when data integrity verifications are performed;
  - smart contracts that allow integrity verification and authorisation for controlled metadata insertion into the DLT;
  - procedures to detect and restore health-data integrity using validation and backup mechanisms to ensure the authentic data is used;
  - security threat models that guided the architecture design to identify, understand, and mitigate potential security risks and vulnerabilities of the proposed architecture.
- demonstration - This step illustrates how the developed artifact operates in a suitable context to address the identified problem. Section "Demonstration" presents the demonstration of

the architecture within the real-world clinical-trial setting of the FAITH project. This scenario details how the architecture, metadata model, and smart contracts mechanisms function cohesively within healthcare workflows to validate the integrity of exchanged data;

- evaluation - The evaluation step involves assessing how effectively the artifact supports a solution to the defined problem. Based on the evaluation, the researcher can decide whether to iterate back to the design phase to improve the artifact or proceed to the final step if the results are satisfactory.
- Following the DSR guidelines, this work uses an "ex post evaluation" [30] meaning that the artifact evaluation is performed after it has been fully designed and implemented. Different DSR evaluation methods exist in the literature [31]. For the architecture proposed in this study, the following evaluation methods were applied [32], namely:
  - illustrative scenario - Artifact can be applied to a synthetic or real-world situation to prove its suitability or utility. The demonstration in Section "Health scenario and demonstration workflow" also serves as this evaluation, as it shows the architecture's usage and suitability in a real healthcare workflow;
  - prototype - Implementation of an artifact aimed to demonstrate the utility or suitability of the artifact. Section "Demonstration environment" presents the demonstration environment, which validates the technical environment used to validate the prototype's correct operation, integration and deployment;
  - technical experiment - This evaluation method verifies the technical performance of the implemented artifact under controlled conditions. It may be use either real-world or synthetic data. In this work, the technical experiment includes integrity-validation tests and the evaluation of system performance based on time-series and hardware metrics on resource-constrained devices. These results are presented in Sections "Technical experiment - integrity evaluation" and "Technical experiment - performance experiment", and an overall discussion is made in Section "Discussion and alignment with the global objective"
- communication - The final step involves communicating the research findings to relevant scientific and professional audiences. This manuscript constitutes the formal communication of the artifact, the methodology, demonstration details, evaluation results, and contributions to both scientific and professional communities.

The presented methodology follows a cyclical process aimed to create useful and effective solutions that can be applied in practice. The next section is dedicated to presenting the artifact design.

Proposed solution - architecture design

In this section, the general topics for a DLT-based approach to improve the security of EHRs are presented. These include an architecture, a health data structure, and smart contracts used to validate and maintain health data while ensuring patient privacy and enable the required authorizations.

System architecture

The architecture presented in Fig. 2 represents the internal structure and inter-communications of the proposed system, showing the data flow, the roles of key system entities, and the necessary modules used to improve the security of an healthcare system.

The proposed architecture in Fig. 2 is composed of two primary layers: the healthcare layer represented by blue color, which includes components executed within the healthcare environment such as internal databases and health services, and the DLT layer, represented by red color, responsible for security services and inter-node connections. In the architecture, purple boxes represent security components that enforce integrity, traceability, and authentication and authorization mechanisms; white modules denote system functions that interact with the DLT layer, enabling integration and utilization of its services; and the green box represents the data source metadata, which is stored on the DLT ledger to provide verifiable information about the data. The numbered steps in the architecture diagram represent the data flow, where steps 1 to 4 correspond to the creation and storage of health data, while steps 5 to 9 illustrate the data retrieval process, including the scenario in which a data integrity breach is detected and handled accordingly.

The architecture begins with the health data sources—also acting as data owners—within a healthcare environment, which continuously generate and provide data to the system. Before this data is used by healthcare services, it is first stored in the system's primary database and simultaneously in a backup database. The backup serves as a reliable safeguard, ensuring data availability in the event of corruption, loss, or unauthorized modifications. To detect such integrity issues, a resource metadata structure is used, which contains information that enables the system to verify whether the original data has been altered. A detailed description of this metadata structure is provided in the next sub-section.

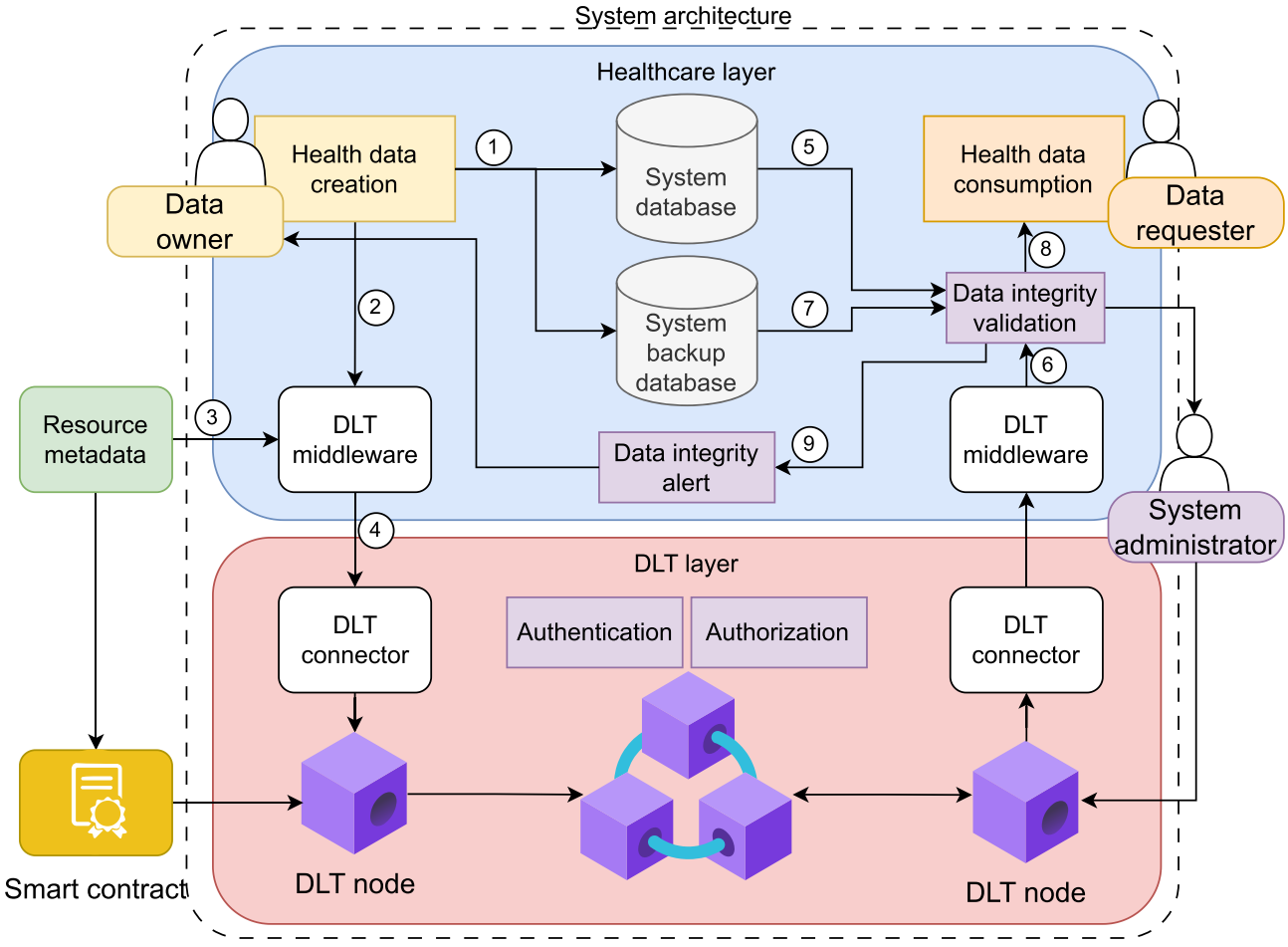

Fig. 2 System architecture applied in an healthcare environment

To facilitate seamless interaction between the healthcare system and the DLT infrastructure, a dedicated DLT connector module can be utilized. This connector abstracts the complexity of interacting with the DLT—such as managing smart contract addresses, metadata field types, and node configurations—thereby simplifying the submission of transactions to the ledger. Moreover, when multiple systems require DLT functionalities, this connector acts as a unified gateway, streamlining secure and consistent access to DLT-based services.

On the DLT layer, both authentication and authorization security mechanisms can be implemented to protect the system. For authentication, each user must possess a cryptographic key pair to interact with the DLT network. This ensures that all actions performed on the ledger are securely attributed to identifiable and verifiable entities. In terms of authorization, smart contracts play a key role by enforcing rules about who is permitted to perform certain actions. Ensuring the information stored in the ledger is trustworthy and reliable is essential not only for maintaining the integrity of system operations, but also for enabling effective data auditing and ensuring compliance with data protection regulations such as the GDPR.

Resource metadata of the healthcare data

In healthcare, data resources can be represented and managed in various ways to ensure effective utilization, accessibility, and governance that are essential to improve patient care, conduct research or making informed healthcare decisions. Those resources include medical expenses data, medical imaging data, pathological parameter data, laboratory test data, etc.

To manage and use the data resource in the proposed architecture, a structured representation was created to ensure proper governance and security within an healthcare system. The defined metadata fields for each resource in this work include:

- resource type (string) - identify the type of resource. It includes questionnaire, laboratory test results, or images. This categorization is crucial for understanding, organizing, and efficiently managing the resources as it allows to group the data based on

shared characteristics, enabling easier retrieval and interpretation;
- sensitive data (boolean) - indicates whether the data is considered sensitive, allowing the system administrator to implement appropriate procedures to protect it and prevent access by unauthorized parties. Examples of sensitive data include salary information, health diagnoses, and insurance records. Additionally, this parameter can be used to determine whether the original resource and its backup must be deleted, particularly in compliance with data protection regulations;
- hash (bytes) - fixed-size value is generated using a hash function based on the resource's data bytes. Since health data can exist in various formats, such as textual data like diagnostic notes or binary data such as medical images, the data resource input is converted to bytes to ensure the compatibility and adaptability of any health data format;
- hashAlgorithm (string) - algorithm that was utilized to generate the hash value. Its deterministic characteristic ensures that the same input always produces the same hash value, making it reliable for tasks like data integrity verification;
- resource version (string) - this field can be used to keep track of changes made over time to a specific resource. As an example, two hospitals might use different versions of HL7, resulting in variations in the resource's properties;
- resource identification (string) - reference that indicates where the original data associated with the resource is stored within the system;
- timestamp (integer) - provides context, relevance, and traceability by recording the time and date when a specific resource was created. This information can be utilized to control access to the resource based on time-related policies, such as limiting access to a two-year period, in compliance with GDPR requirements.

Considering sensitive data itself is not stored within the resource metadata, data privacy is preserved.

Smart contract design

Smart contract functionality guarantees that information available on the DLT is authentic and associated with this work. It is divided into two parts. The first defines the structural part of the resource information and it contains the resource fields described in previous subsection together with their specific types, e.g., resource version only allows string values. The second part of the smart contract includes a set of authorization constraints to enforce access control over metadata insertion. This contract restricts the creation of metadata entries to only those identities that have been authorized by a system administrator, ensuring that only trusted sources can submit integrity-related information to the DLT. Three different authorizations were defined in the smart contract: to ensure that only correct data are available on the DLT, an authorized list of users who can create the resource information can be specified; only the smart contract owner is allowed to add or remove the entities that are authorized to create resource metadata; only the smart owner is allowed to change the address which allows or disables users to insert new resources into the DLT. The authorization mechanism is critical in healthcare and industrial environments because it ensures that only legitimate data owners can create the resource metadata, thereby guaranteeing that the data hash values used to verify the original data remain authentic.

Data integrity verification and restoration procedures

As noted in [33], the inherent immutability of DLT renders data stored on-chain permanent, which poses challenges for systems that use this technology and must comply with privacy regulation such as GDPR. To prevent user privacy breaches and mitigate scalability issues associated with storing data directly on the DLT, only metadata described in Section "Resource metadata of the healthcare data" and pointers to externally stored raw data are recorded on-chain. Although metadata is shared among all DLT nodes, it contains no personal identifiers or attributes that could reveal user's identity. As a result, if a health resource must be deleted within the healthcare system, it can be removed permanently without violating GDPR obligations, since the corresponding ledger entry contains no sensitive or identifiable data.

To ensure reliable system behaviour, it is essential to address data protection from the earliest stages of the data. As noted in the literature [34], to use a hash to verify the integrity of data, it should be calculated at the moment and on the same device where the data were originally generated or last modified. Considering this fact, this work employs a MDC approach, whereby a cryptographic hash of the data is generated for integrity verification when the data are used. To strengthen this method, the reference hash is stored in a DLT system, ensuring that the integrity check itself cannot be compromised. Considering this, the proposed approach operates in two stages: data creation and second, when the data are used.

Data creation

To safeguard against data compromise and ensure its recovery, the data is stored both in the system's primary database and in an independent backup storage. The purpose of this backup storage is to provide a secure and reliable copy of the original data, to ensure integrity in the

event of corruption, data loss, or system errors. Afterward, a hash algorithm is applied to the data to generate a unique hash value, which is included in the resource information emitted to the DLT. This hash value allows the detection of any changes in the data, even a single byte, as any modification will result in a different hash value. Such a change can trigger an alert to inform users of a possible unintentional data alteration. Since all DLT nodes share the same information, the recorded hash value is visible to all participants in the network, enabling integrity checks without exposing the data or compromising the privacy of the data owner.

Data retrieval

Data integrity validation is triggered when the DR initiates a retrieval request, ensuring that the integrity of the data is verified before it is accessed from storage. In this stage, the system first retrieves the data and then recalculates its hash value using the same algorithm. This newly generated hash value is then compared with the corresponding hash value stored on the DLT. If they match exactly, it confirms that the data have not been tampered with and that the information is provided to the user. If an integrity violation is detected, a notification is triggered to both the data owner and the system administrator, following GDPR guidelines. Additionally, an incident report must be created, potential additional breaches should be investigated using DLT records, and the compromised data in the system database should be replaced with the correct version from the backup.

Threat model

The requirement for a trusted environment is an health environment, and security assumptions were considered during the design of the proposed DLT-based health-data integrity mechanism presented in this work. Considering the importance of maintaining GDPR-compliant and privacy-preserving data handling, this study considered three threat scenarios to help identify the inherent risks that may be introduced into the system [35].

The first identified threat involves server-side malicious tampering, where an attacker with access to the healthcare institution's database may compromise the EHR stored off-chain. This threat is mitigated by registering the EHRs hash value in the immutable ledger of the DLT at creation time and validating its integrity by comparing the recomputed hash value with the hash stored on-chain.

The second threat is unauthorised modification of metadata, where an adversary may attempt to forge the health metadata. This risk in this scenario is the introduction of falsified health metadata that, if accepted, could mislead the integrity verification process. This threat is mitigated through the authorisation mechanisms enforced by the smart contract that ensures that only approved blockchain addresses are permitted to submit metadata to the DLT.

The third scenario considers insider collusion threat, where a trusted system entity deliberately changes health data from the healthcare database. This work considers and mitigates this type of threat by ensuring that only authorized entities can submit metadata and by relying on DLT immutability. In addition, metadata versioning and the cryptographic signing of each transaction provide auditability and allow the system to detect attempts to register manipulated or inconsistent metadata.

For the demonstration of the mentioned threat scenarios, this work made several assumptions that define the security boundaries within which the proposed architecture is expected to operate. First, we assumed that a supermajority of the validator nodes behave honestly, which is essential to ensure that the DLT data cannot be adultered. Second, the model assumes that the security administrator is a fully trusted authority responsible to manage DLT authorizations and to ensure that only approved system entities are permitted to insert data on-chain. Finally, the system assumes that data is correct at the moment of metadata creation, meaning earlier corruption cannot be detected. These assumptions enable tamper-evident and GDPR-compliant verification but also represent weaknesses and trade-offs of the proposed approach.

Demonstration

This section demonstrates how the proposed DLT-based architecture operates in practice within the real healthcare clinical-trial context of the FAITH project. The demonstration corresponds to the DSR demonstration step and exemplifies the artifact's functionality in a realistic setting, integrating the healthcare workflow, the system's technical infrastructure, its participating entities and the integrity-verification mechanisms. It also serves as the basis for the illustrative scenario evaluation method from Section "Illustrative scenario evaluation".

Health scenario and demonstration workflow

The demonstration scenario reflected the data workflow of the FAITH clinical trials. The FAITH project was dedicated to the design of a platform to promote mental health awareness and provide intelligent support for post-cancer care, helping patients improve their quality of life. It resulted in a platform that collects EHRs from the patient's mobile phone, which are then analysed by healthcare professionals to determine the need for clinical intervention.

The duration of the trial was one year and a half, during which a total of 200 patients were enrolled. Each patient could submit up to 20 entries per day, all of which were

stored on the same healthcare server responsible for maintaining the EHR. Three types of participants were involved in this scenario, as shown in Fig. 3: patient, doctor, and security administrator. The demonstration workflow comprised two steps: when data are created and when the data are used.

Data creation and metadata submission

What a patient contributes in terms of data creation, and the subsequent submission of its metadata, is explained in the following steps:

- configuration phase - The security administrator configured the smart contract authorisation rules on the DLT, to ensure that only trusted entities were permitted to submit metadata;
- resource creation - The patient generated a health resource, such as a questionnaire responses or clinical measurements;
- metadata generation - After the storage of the resource was stored, the healthcare system generated the corresponding metadata, including resource type, hash value, sensitivity level, version, and timestamp, and submitted it to the DLT through the authorized smart-contract interface;
- DLT processing - The metadata was transmitted to the DLT Middleware, which handled the interaction with the smart contract and stored the metadata on the ledger.

Data access and integrity verification

When the doctor or patient later accessed a stored health resource, the process of data integrity verification was initiated, which consisted in the following steps:

- resource retrieval - The patient or doctor accessed the health resource from the healthcare system database;
- hash computation - The system recomputed the hash value of the retrieved data;
- metadata retrieval - The healthcare system retrieved the corresponding resource metadata from the local DLT node;
- integrity validation - The authenticity of the health data was verified by the Data Integrity Validation by comparing the recomputed hash value;
- integrity outcome
  - if the values matched, the data was confirmed to be intact and was safely delivered to the data requester;
  - if the values differed, the system triggered an alert to both data owner and the security administrator, who investigated the breach and restored the correct data from the backup database.

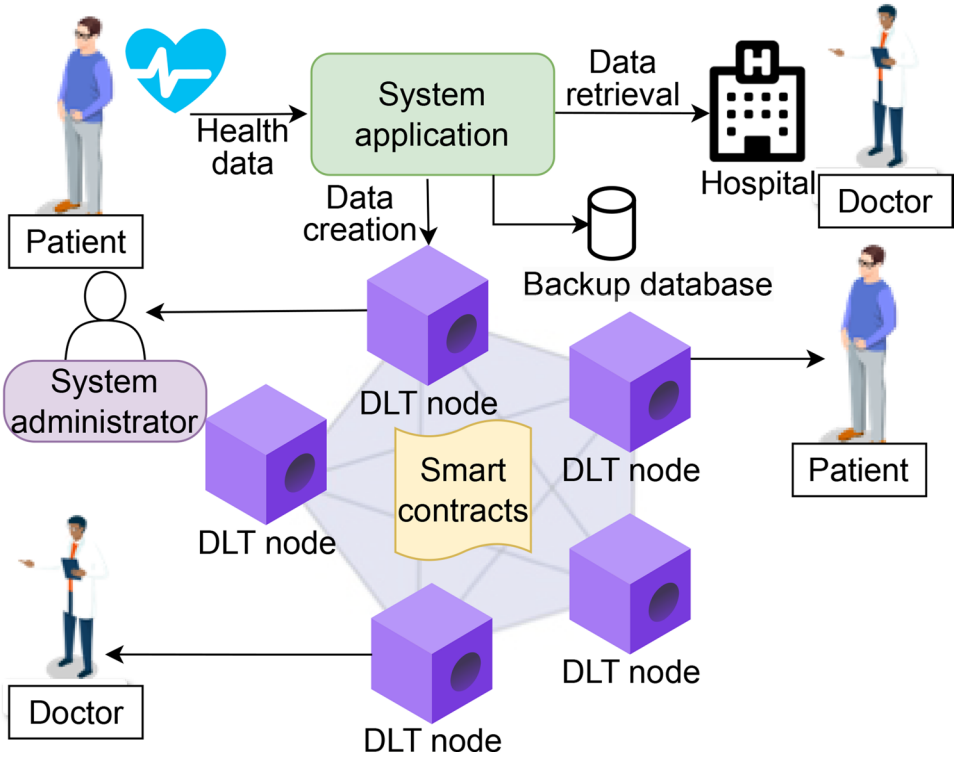

Fig. 3 Healthcare scenario integrated with a DLT

This scenario, implemented during FAITH Project, demonstrated how the proposed architecture coordinated security mechanisms, enforced GDPR-compliant processing, and provided tamper-evident verification within a real healthcare workflow.

Demonstration environment

The demonstration was conducted using a full prototype deployment that replicated the technological conditions of the FAITH clinical-trial environment. Figure 4 illustrates the physical and logical deployment used during the demonstration, including the distributed DLT nodes, hospital systems, backup repositories, and the administrator environment.

The prototype consisted of a private Ethereum blockchain to ensure that only authorized parties could access the ledger data and allow the creation of unlimited transactions into the DLT without monetary restrictions. The private Ethereum network also allowed us to have a better network configuration and define how the transactions were created and validated. It used the Proof of Authority (PoA) consensus mechanism (also known as Clique) [36] as it allows to define which nodes can validate the transactions (the healthcare and system administration were part of this validation process).

In this DLT network, new blocks of the blockchain were created only when a transaction was submitted, rather than at fixed time intervals. This approach improves efficiency by eliminating empty blocks, reducing latency as transactions do not need to wait for a predetermined time or block size before processing, and contributing to economic efficiency by minimizing the waste of computational resources and energy on creating empty or underutilized blocks. All these characteristics were defined on the first block of the Ethereum blockchain, referred to as the genesis block. Each node was initialized using the same genesis file to ensure a consistent network state. After network initialization, After network initialization, the system administrator deployed the smart contract to the blockchain, establishing the contract owner and the blockchain addresses authorized to create metadata entries. Each node maintained a synchronized copy of the ledger, ensuring tamper-evidence and traceability. In this PoA setup, three validator nodes, represented by blue color cubes in Fig. 4, were responsible to validate the transactions and create new blocks in the blockchain.

As illustrated in Fig. 4, the system was composed of six Ethereum blockchain independent nodes in geographically separated locations, four of which were Raspberry Pi 4 Model B single board computers with 8 GB LPDDR4 RAM and a 64-bit quad-core Cortex-A72 processor. All devices ran Debian GNU/Linux 12 with Go Ethereum v1.13.4.

The healthcare system modules included two hospitals, each with its own private database for storing EHRs, as well as a shared backup database used to restore data in case of integrity breaches. Each hospital environment also operated an Ethereum node to enable local metadata retrieval and validation.

The administrator management module also operated its own Ethereum node to manage smart contract permissions, enabling it to assign authorisation rights for metadata creation. The entire system, from the transactions logic to the writing to the blockchain, is implemented using JavaScript language due to the simplicity of the language and the availability of libraries. Such libraries, for example, the Web.js version 4.2.1, provided the necessary tools and functionalities to interact with the Ethereum blockchain as well as their smart contracts.

This deployment environment represented the operational conditions of the FAITH platform and enabled the execution of the end-to-end workflow described for the demonstration of this study.

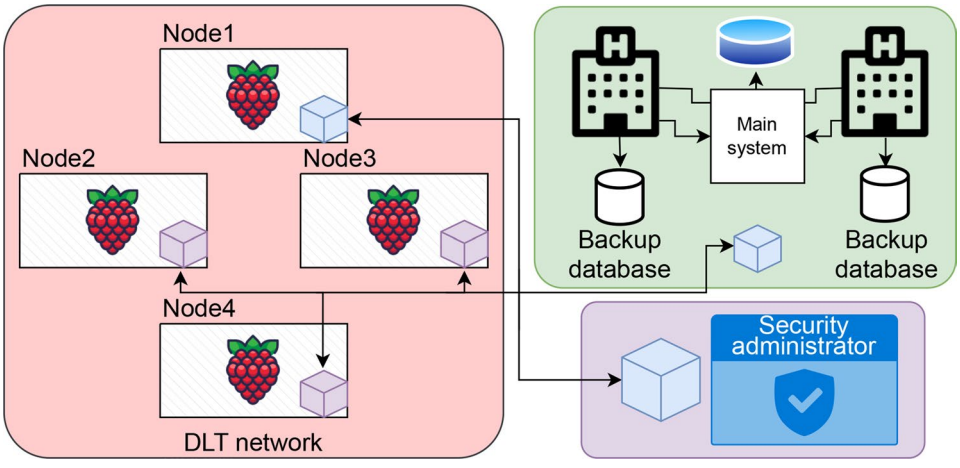

Fig. 4 Deployed prototype of the demonstration environment with Raspberry Pi nodes to validate healthcare data integrity

Evaluation

Evaluation is an important activity in the DSR methodology, as it allows to confirm that utility, feasibility, and efficiency of the artifact. This section presents two evaluation methods comprising three evaluation procedures through which the proposed architecture was assessed to determine whether it adequately addresses the problems identified in this work.

Illustrative scenario evaluation

The suitability and utility of the artifact were evaluated using the illustrative scenario method, by integrating the solution into a realistic healthcare context. This evaluation also corresponds to the demonstration scenario presented in Section "Health scenario and demonstration workflow", where the architecture was deployed and executed within an health clinical trial which lasted 18 months. Specifically, the scenario provided evidence of the following:

- prototype deployment in a healthcare environment - All architectural components were implemented and deployed within FAITH platform, which allowed the clinical trials to utilize the functionalities proposed in this work;
- health metadata - Whenever a health resource was created, the system generated a corresponding metadata information containing the fields required for subsequent verification;
- DLT integration - The utilized blockchain allowed to stored metadata on the ledger, benefiting from the on-chain immutability typically provided by a DLT.
- privacy-compliance - Considering the traditional characteristics of DLT are not compliant with GDPR, namely immutability and broad data availability for all participant nodes, only the metadata was stored on the ledger. This ensured privacy-preserving integrity verification since the on-chain data does not contain direct patient identifiers and had insufficient information to infer the patient identity;
- integrity verification - When patients or doctors retrieved health data, the system recomputed the data hash value and compared it with the retrieved hash stored on-chain, thus enabling trusted integrity verification;
- breach detection and recovery - When discrepancies were identified, the system automatically triggered alerts and restored the correct data version from the health backup repository, preventing the use of compromised data.

The behaviour identified throughout the scenario confirms that the artifact operated as designed, fulfilling the objective defined in Section "Methodology". It supported GDPR-compliant processing, preserved data integrity, and seamlessly integrated a real healthcare workflow. Importantly, the scenario validated that the metadata-based approach enabled the system to detect manipulation attempts using cryptographic mechanisms, without exposing sensitive patient information on the ledger.

Technical experiment - integrity evaluation

The technical performance, validity, and efficiency of the artifact in DSR can be evaluated through a Technical Experiment. This section evaluates the capability of the proposed solution to identify a data integrity adulteration using the mechanisms defined in the architecture.

Figure 5 illustrates the evaluation flow described in this section, showing two examples: a correctly validated case (represented by blue arrow), and a manipulated case (represented by red arrow), where the altered data was identified as non-original data.

To evaluate integrity preservation, the experiment was conducted using two different scenarios:

Scenario 1—Original data (valid integrity)

In first scenario, the data retrieved from the health database was unmodified. When the system recomputed the hash value of this data and compared it, both values matched. Considering the metadata and the recomputed hash were identical, the system validated the integrity of the resource and the data was retrieved to the doctor.

Scenario 2—Modified data (integrity breach)

In the second scenario, one value of the patient's health data (e.g., glucose level) was intentionally modified and compared it with the immutable hash stored on-chain, to simulate data tampering. Considered the hash values were different, the system automatically identified a mismatch between the recomputed and immutable hash stored on-chain, enabling the system to consistently identify the integrity breach. The modified EHR was flagged as compromised, preventing its use in clinical decision-making.

Figure 5 clearly shows the importance of using cryptographic mechanisms to detect data integrity changes. Despite having similar structure and fields, its hash function produces a distinct hash value, identifying a data alteration.

Overall, this experiment confirms that the proposed mechanism reliably detects integrity breaches while maintaining GDPR-compliant handling of metadata.

Technical experiment - performance experiment

The Performance Experiment evaluates the technical behaviour of the implemented artifact by examining how the system performs under real workload conditions on resource-constrained devices. The performance

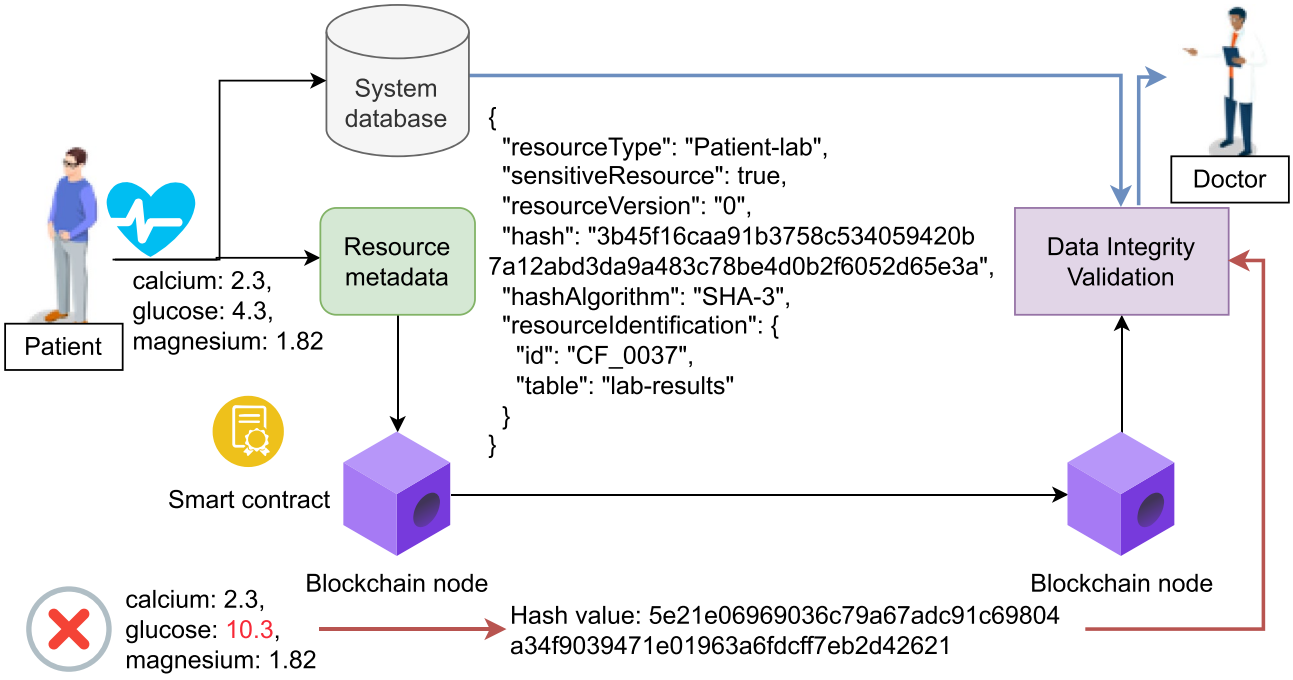

Fig. 5 Integrity evaluation. Resource metadata is shared with all blockchain nodes and its information is used to verify the integrity of patient health data

experiment evaluation is organized into two parts: the first describes the measurement protocol used to collect the time-series and hardware metrics, and the second presents the analysis of these measurements to assess the system's performance.

Measurements protocol

To evaluate the consequences of integrating the proposed architecture into an healthcare system, two distinct sets of metrics were collected. The first set aims to infer the response time required to create a resource information within a DLT. These metrics were acquired on the components represented Fig. 6. The second set focuses on hardware evaluation, on the Raspberry Pi DLT nodes represented in Fig. 4. Both these measurements were performed within the same experiment.

To ensure the validity of time and hardware measurements, all experiments presented in this section were conducted in a controlled environment in which every device was configured with the same operating system and the same hardware specifications. All Raspberry Pis operated with independent and stable power supplies, and thermal and environmental conditions were kept constant to reduce confounding factors. Moreover, all Raspberry Pis contained the same libraries, background processes, and both Bluetooth, Wi-Fi and graphical interface were disabled. These devices nodes were connected to the same local network without competing traffic, ensuring consistent latency and network behaviour across the experiments. Each system component processed the same workload of 5000 requests, sent at 500 ms intervals, where each request contained a JSON structure whose internal fields varied for every submission to reflect realistic data-generation conditions. Only the raw hardware measurements and request timestamps were stored, to avoid additional processing or filtering steps that could distort the results or introduce computational overhead.

Response time metrics measurements The time series data collection was performed in laboratory conditions, i.e., the computer only had installed the operating system, the required libraries (e.g., Node JS and Go), and the components that were measured, which are represented in Fig. 6. The computer was a MacBook with macOS 12.7.3 operating system with a quad-core Intel Core i7 processor and 16 GB of memory RAM. This computer had a regular node and it was connected with all nodes represented in Fig. 4. A script was created to emulate the REST requests made by the hospital and the timestamps before the request and after the response were stored, that correspond to the numbers 1 and 6 of Fig. 6, respectively. The ratio between these values corresponds to the total time to call the health function, which includes its main functionalities (e.g., storing data on the hospital database and on the backup database, hospital data analytics, etc) as well as submitting a resource on the DLT. It used the relational database Maria DB for the backup database. The second group of timestamps had the purpose of registering the time it takes to create the necessary information to

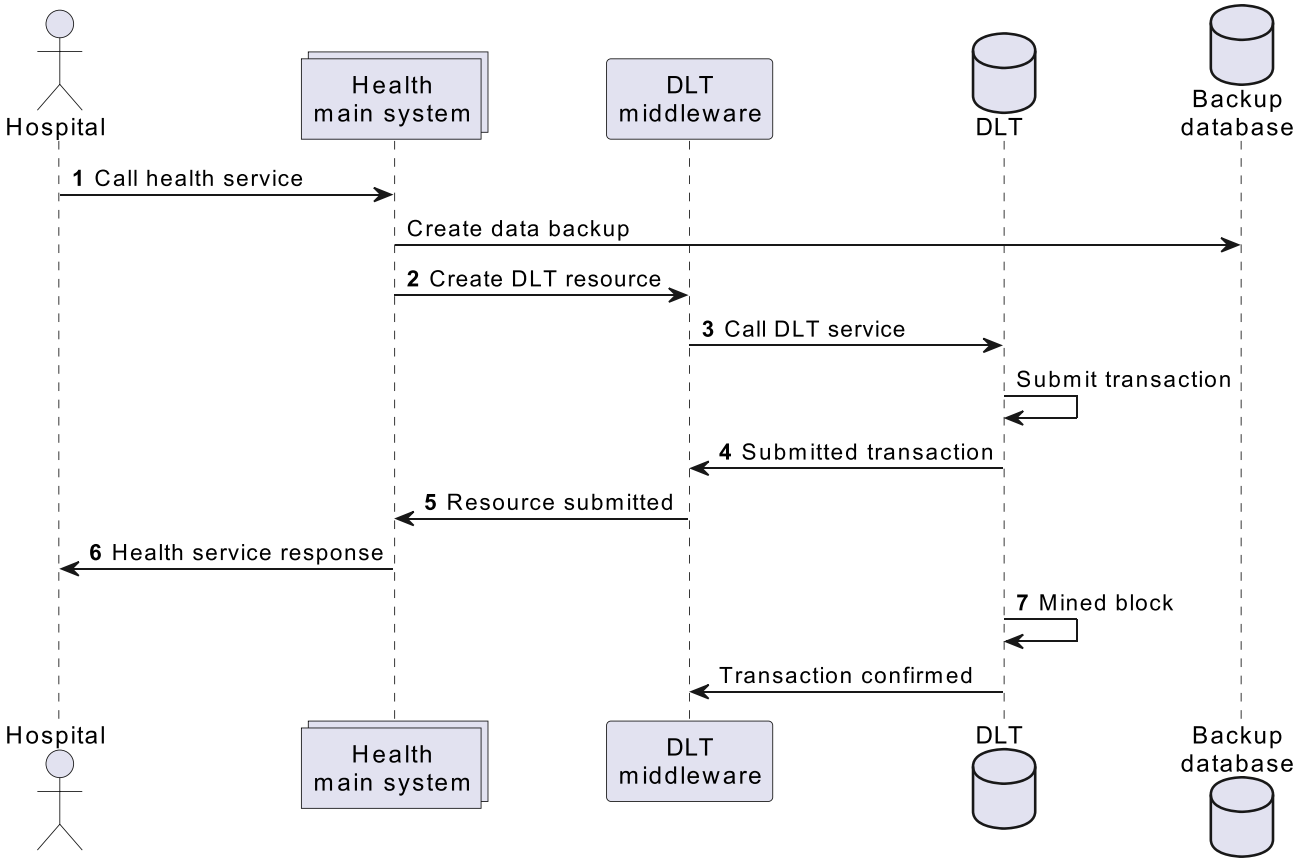

Fig. 6 Time series measurements data flow

submit a resource to the DLT. For this reason, two timestamps were registered on this component, before parsing the data to create the resource and after the response from the DLT middleware component, which correspond to the numbers 2 and 5 of Fig. 6, respectively. The ratio of the timestamps from this component corresponds to the time it takes to create resource information and submit to the DLT. The component DLT middleware also recorded two timestamps which had the purpose to measure the time it takes to submit a resource on the DLT. These timestamps were registered before submitting a transaction to the DLT and after receiving the information of a pending transaction to be appended to the DLT and correspond to the numbers 3 and 4 of Fig. 6. The purpose of this last ratio is to measure the time it takes to submit a resource into the DLT. There is a gap between the time when a transition is submitted to the DLT and the time when it is confirmed, which is represented by the 'Mined block'in Fig. 6 and corresponds to the number 7. This value was calculated based on the difference between the block creation time and the submission transaction time, which was less than one second for all transactions. Since this value is not relevant to this work, it will not be discussed further.

Hardware measurements The subsequent performance measurements were conducted with a DLT configuration closely resembling that employed in the FAITH project. To access the resource bottleneck of DLT usage on the system using the metadata structure, metrics were collected from three Raspberry Pi. These devices shared the same hardware and identical software characteristics but served different roles in the system: one operated as a validator node, another as a regular node, and the third without any DLT software. Before conducting the tests, the Raspberry Pis executed two bash scripts: one responsible for storing the disk I/O measurements and another that stored, in memory, the raw data related to CPU temperature, memory RAM usage, CPU usage and network traffic. All measurements were stored in CSV files. To minimize disk I/O and reduce resource usage, preventing interference with the performance measurements on the Raspberry Pis, the last CSV file stored 100 measures each time. Furthermore, the CSV files only contained the values that were used to compute the measurements, i.e., the raw values. The algorithms for computing the performance measurements were exclusively executed during data analysis, utilizing an R script to minimize resource utilization on the DLT nodes. Considering the update frequency and precision of some Raspberry Pi measurements are limited by the

underlying hardware and kernel, and acquiring accurate time measures involves relative resource-intensive system calls which influence the acquired data for this work, all metrics were made at the level of one second and on each Raspberry Pi itself.

The Raspberry Pi temperature was acquired from `/sys/class/thermal/thermal_zone0/temp` file, which contains the CPU temperature in millidegrees Celsius. The data retrieved from this file was converted to degrees Celsius, with precision up to two decimal places. The "free" command is a Linux command that was used to get the memory RAM information at the time of execution. It provides the total installed RAM, RAM used by the system as well as unused RAM. The overall RAM usage percentage of the Raspberry Pi was calculated with Eq. 1.

$$\mathrm{RAM}[t] = \frac{\mathrm{RAM}_{used}[t]}{\mathrm{RAM}_{Total}[t]} \times 100. \quad (1)$$

On a Raspberry Pi, the data related to processor usage can be read from the `/proc/stat` file. It contains one or more rows, depending on the number of processors the system has. The very first line, labelled "cpu", aggregates the numbers in all other "cpuN" lines. This was the line which was used to compute the CPU usage for this work. The numbers provided in this file represent the cumulative time the CPU has spent performing the different modes of execution of a CPU since the system's initial boot. The basic modes identify the amount of time the CPU was busy (user, nice and system processes), in idle state, interrupted and some other states. As the provided values from this file are accumulatively progressing over time, the CPU usage must be determined by computing the variance between the counter values. Thus, the CPU processor usage percentage is computed according to Eq. 2.

$$\mathrm{CPU}[t] = \frac{\mathrm{CPU}_{busy}[t] - \mathrm{CPU}_{busy}[t-1]}{\mathrm{CPU}_{total}[t] - \mathrm{CPU}_{total}[t-1]} \times 100. \quad (2)$$

The CPU usage is calculated by dividing the busy amount of time CPU_{busy} by the total number of time CPU_{total}.

The network activity is another important characteristic of DLT since it refers to the communication and data transfer processes between nodes within the network. It encompasses both data received ($\mathrm{ETH}_{\mathrm{RX}}$) and data transmitted ($\mathrm{ETH}_{\mathrm{TX}}$), and in a Linux system and using an ethernet cable, they can be acquired through the files `/sys/class/net/eth0/statistics/rx_bytes` and `/sys/class/net/eth0/statistics/tx_bytes`, respectively.

DLT nodes usually rely on read-and-write operations to the disk. To determine the capability of the node's storage, the transactions per second (TPS) were measured, which refers to the number of input/output - $\mathrm{I/O}_{\mathrm{TPS}}$ operations completed by a storage device. This parameter was acquired by using the 'iostat'Linux command, which provides system's I/O statistics for the system's devices and partitions. Since this command needs to be running continuously in order to acquire the real-time I/O values, this command was executed in a separate script.

Analysis of results

To investigate the performance of the proposed DLT-based design, time and resource consumptions measurements were performed. It was conducted 5000 requests in 500 ms interval and all measurements were executed during the same experiment. To enable the comparison between measurements, all requests were stored with a unique ID, which was defined at the hospital request side and acquired by the remaining components through parsing the request body.

Figure 7 shows the time measurements scatter plot for the hospital requests, health system and DLT middleware, from the data flow presented in Fig. 6 with colours purple, yellow and blue, respectively. All times were determined by calculating the difference between the end and the start of a request and associating them with their corresponding request number. Even though the hospital request used a database to store data, the request time is identical to the time it took to perform the hospital request. The mean time was 16.94 ms for the hospital and 14.47 ms for the health system requests. The DLT middleware component had a mean time of 1051 ms, which might be influenced by the time the web3.js library uses to create and submit a transaction to the Ethereum blockchain. Multiple ledger accesses to create and send a new transaction to the ledger are performed by this library, such as to determine the transaction gas price or the account's pending transaction number. Similar to a database, each disk access takes time, which affects the overall performance of the service. However, for the proposed scenario, a 1051 ms delay does not affect the health system.

To provide a clear baseline for comparison, the evaluation utilized three Raspberry Pi setups: a validator node, a non-validator node, and a baseline device without any blockchain activity. The baseline served as a control condition, enabling us to isolate the computational overhead introduced solely by running a DLT node and, in the case of the validator, the additional processing required to validate the metadata entries created in this study. Figure 8 presents six hardware measurements collected across these three configurations, where the validator is shown in yellow, the non-validator in purple, and the baseline

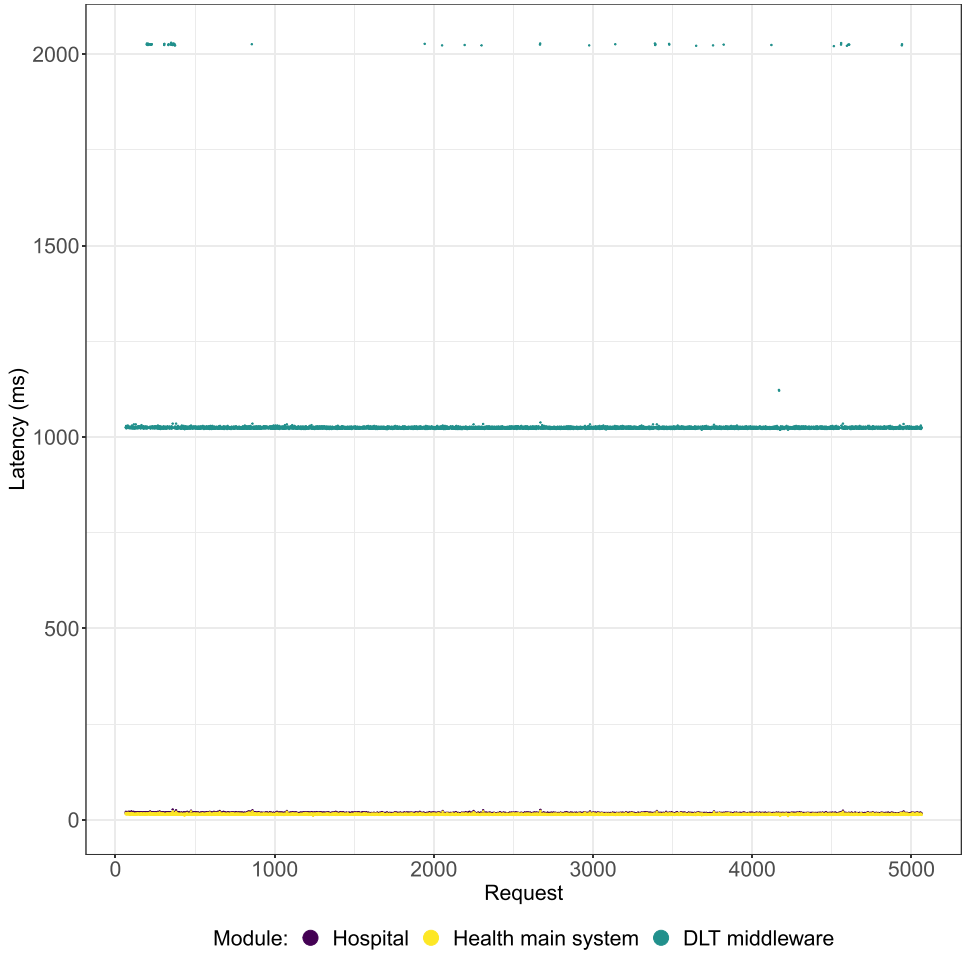

Fig. 7 Time measurements of health components integrated with a DLT

Raspberry Pi in blue, which does not execute any DLT processes.

To better visualize the patterns of the scatterplot from Fig. 8, the "loess" method of the "geom_smooth" function from the ggplot2 package was used. One-way analyses of variance (one-way ANOVAs) were performed to verify if there were statistically significant differences among the three Raspberry Pi devices, for each measurement. The data were all aggregated into one data frame in R, and the measurements were assigned between three groups that correspond to each device: validator, non-validator and baseline. For pairwise comparisons among Raspberry Pi devices, post hoc Tukey honest significant difference (HSD) tests were used.

One-way ANOVAs results showed statistically significant differences among devices for all measurements (Temperature: $F(2, 7642) = 3359.74$, $p < 0.001$; RAM: $F(2, 7642) = 2788810.90$, $p < 0.001$; CPU: $F(2, 7642) = 14.53$, $p < 0.001$; ($\mathrm{ETH_{RX}}$): $F(2, 7642) = 5957.76$, $p < 0.001$; ($\mathrm{ETH_{TX}}$): $F(2, 7642) = 10036.24$, $p < 0.001$; $\mathrm{I/O_{TPS}}$: $F(2, 7771) = 13324.98$, $p < 0.001$).

The Tukey's HSD post hoc test for device temperature revealed significant differences among all groups (adjusted $p < 0.001$), with baseline and validator showing the lowest and highest temperatures, respectively (Fig. 8). These findings suggest a direct influence of the Ethereum DLT on temperature across all devices, with mean differences of 1.44 °C and 0.45 °C between validator with baseline and validator with non-validator, respectively. Similarly, RAM Tukey's HSD post hoc test results indicated significant differences among all groups ($p < 0.001$), with baseline showing notably lower values (Fig. 8). Regarding the CPU measurements, significant differences were observed between the baseline and validator and non-validator devices ($p < 0.001$). Both $\mathrm{ETH_{RX}}$ and $\mathrm{ETH_{RX}}$ network activities showed significant differences among all groups ($p < 0.001$), with the highest mean differences observed between the baseline (lower values) and the validator (higher value) and non-validator nodes. This network behaviour is attributed to constant communication among nodes to verify if new information has been approved and added to the ledger. Regarding $\mathrm{I/O_{TPS}}$ measurement, Turkey's HSD post hoc test

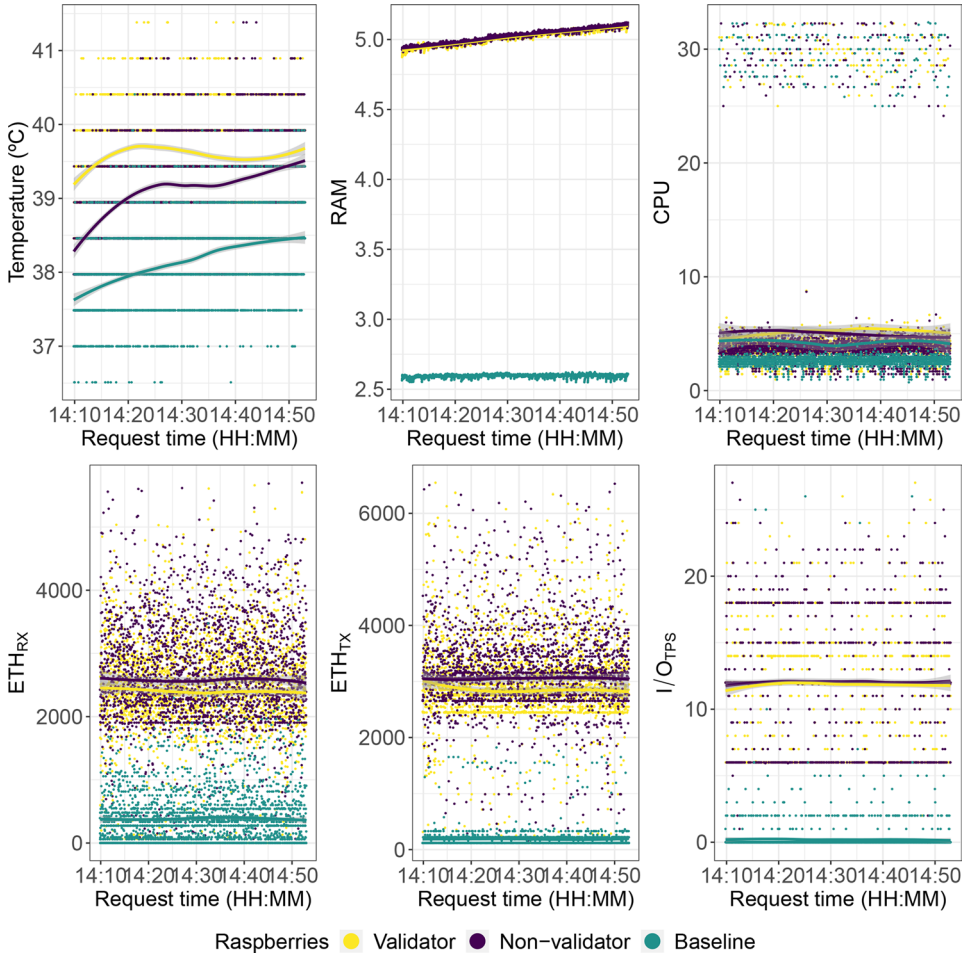

Fig. 8 Raspberry Pi hardware measurements for each type of node

revealed a significant difference between the baseline and the other devices ($p < 0.001$). This result can be attributed to storage disk operations occurring only when a new block is added to the ledger by both ledger nodes devices.

Discussion and alignment with the global objective

The results obtained across the illustrative scenario (Section "Illustrative scenario evaluation") and technical experiments (Sections "Technical experiment - integrity evaluation" and "Technical experiment - performance experiment") demonstrate that the proposed artifact successfully fulfills the defined objective.

First, the integrity-verification mechanism was validated through a technical experiment on Section "Technical experiment - integrity evaluation". The prototype consistently detected the data tampering attempts by comparing the recomputed hash values with the immutable information stored on the DLT. This confirms that the artifact in fact enables the verification and validation of exchanged data integrity.

Second, the design decision to use the health resource metadata ensured privacy preservation and GDPR compliance, even with the utilization of DLT. As shown in Sections "Illustrative scenario evaluation" and "Technical experiment - integrity evaluation", the metadata stored on-chain contained no personal identifiers and no attributes capable of disclosing patient identity. The immutable information stored on the DLT do not affect the compliance with GDPR requirements when an health resource is requested to be deleted. This demonstrates that the proposed mechanism supports privacy-preserving and regulation-compliant processing.

Third, the operational compatibility with healthcare environments was demonstrated through the FAITH clinical-trial integration (Section "Illustrative scenario evaluation") and the technical experiments. The developed prototype was successfully utilized in a real clinical workflow involving more than 200 patients and different health entities. Additionally, the performance experiments showed that the system remained stable on resource-constrained hardware. This validates that the architecture can be deployable under realistic healthcare constraints.

Altogether, the demonstration and experimental results provide strong evidence that the proposed artifact extends and improves upon the prior systems identified in Section "Background", by adding a data integrity verification, GDPR-compliant processing, automatic authorisation mechanisms enforced through smart contracts and DLT functionalities. Overall, this study achieves its global objective: it delivers a DLT-based mechanism capable of maintaining data-integrity validation that is secure, verifiable, privacy-preserving, GDPR-compliant, and operationally feasible within real healthcare environments.

Conclusion

In this paper, a DLT architecture to enhance healthcare data integrity was proposed for the storage and sharing of health-related information. A demonstrator using Raspberry Pi 4 nodes has been evaluated and verified a small overhead when applying our solution to a healthcare system, which compensates for the security improvement for the health data and continuous data protection against manipulation attacks. This is crucial when data are connected with individuals and their health. Compliant with the GDPR, this work was used in the FAITH research project to guarantee the authenticity of the health questionnaires and their answers for the hospital's interface. The health resources and their backups were stored in separate databases, on the hospital's premises and the DLT was used to store the resource information and their location. The whole architecture presents the solution needed for an hospital-based data collection system, applicable in research such as the present case with the FAITH project but it is a proved solution for any other hospital circumstance where data integrity and data protection is mandatory.

The legacy of this work also extends such requisites beyond traditional databases to a full secure and auditable DLT based infrastructure. It is therefore concluded that this document serves as a reference for those aiming to design a DLT-based data integrity system to support a secure and trustworthy eHealth data infrastructure, including the use of low-resource microcontrollers, such as Raspberry Pi, as part of the DLT layer. It should be noted the architecture mitigates the threats described in Section "Threat model", however, this study focuses primarily on verifying and ensuring the integrity of health data used within the system. This limitation should be explored in a future work extend the approach beyond detection and toward the prevention of data integrity violations.

Author contributions

J.G., F.F., and J.S. conceived of the presented idea. J.G. and F.F. developed the theoretical analysis. J.S., and R.G. supervised the findings of this work. All authors provided critical feedback and helped shaping the research, discussed the results and contributed to the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding

The application of these results in the YACHAY and EPAI research projects is a reality being designed, aiming once again to ensure the authenticity of the project's information. This solution is departing from lessons learned with the presented implementation, whereas the authentication mechanism will apply the hardware devices supporting the DLT infrastructure along with the validated software, as documented in the present research work.
This work was supported in part by the European Research Council through the European Union's Horizon 2020 Research and Innovation Programme under Grant 875,358 (FAITH project) and in part by the European Research Council through the Erasmus+ program under grants 619,410-EPP-1–2020-1-PE-EPPKA2-CBHE-JP and 101,082,868-ERASMUS-EDU-2022-CBHE-STRAND-2 (YACHAY and EPAI projects).

Data availability

The software supporting the conclusions of this article are available in DLT middleware repository, https://code.grisenergia.pt/faith/dlt-middleware; DLT connector repository, https://code.grisenergia.pt/faith/faith-dlt-connector.

Declarations

Competing interests

The authors declare no competing interests.

Received: 9 June 2025 / Accepted: 9 January 2026
Published online: 05 February 2026

References

1. Esposito C, De Santis A, Tortora G et al (2018) Blockchain: a panacea for healthcare cloud-based data security and privacy? IEEE Cloud Comput 5(1):31–37. https://doi.org/10.1109/mcc.2018.011791712
2. Liu C, Yang C, Zhang X et al (2015) External integrity verification for outsourced big data in cloud and IoT: a big picture. Future Gener Comput Syst 49:58–67. https://doi.org/10.1016/j.future.2014.08.007
3. Hathaliya JJ, Tanwar S (2020) An exhaustive survey on security and privacy issues in healthcare 4.0. Comput Commun 153:311–335. https://doi.org/10.1016/j.comcom.2020.02.018
4. Pandey AK, Khan AI, Abushark YB et al (2020) Key issues in healthcare data integrity: analysis and recommendations. IEEE Access 8:40612–40628. https://doi.org/10.1109/access.2020.2976687
5. Chen YJ, Wang LC, Wang S (2020) Stochastic blockchain for IoT data integrity. IEEE Trans on Network Sci Eng 7(1):373–384. https://doi.org/10.1109/tnse.2018.2887236
6. Wang H, Zhang J (2019) Blockchain based data integrity verification for large-scale IoT data. IEEE Access 7:164996–165006. https://doi.org/10.1109/access.2019.2952635
7. Zia MF, Benbouzid M, Elbouchikhi E et al (2020) Microgrid Transactive energy: review, architectures, distributed ledger technologies, and market analysis. IEEE Access 8:19410–19432. https://doi.org/10.1109/access.2020.2968402
8. European Commission (2020) Federated artificial intelligence solution for moniToring mental health status after cancer treatment. https://cordis.europa.eu/project/id/875358
9. Keshta I, Odeh A (2021) Security and privacy of electronic health records: concerns and challenges. Egypt Inf J 22(2):177–183. https://doi.org/10.1016/j.eij.2020.07.003
10. Christensen AI, Lau CJ, Kristensen PL et al (2022) 35 years of health surveys in Denmark: a backbone of public health practice and research. Scand J Public Health 50(7):914–918. https://doi.org/10.1177/14034948221083113
11. European Parliament and Council of European Union (2016 2016/679) Regulation (EU). https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679%26from=EN
12. Marron J (2024) Implementing the health insurance portability and accountability act (HIPAA) security rule: a cybersecurity resource Guide. NIST Spec Publ (SP) NIST SP 800-66r2. https://doi.org/10.6028/nist.sp.800-66r2
13. van Steen M, Tanenbaum A (2017) Distributed systems, 3rd edn edn. Maarten van Steen

14. Zarour M, Alenezi M, Ansari MTJ et al (2021) Ensuring data integrity of healthcare information in the era of digital health. Healthcare Technol Lett 8(3):66–77. https://doi.org/10.1049/htl2.12008
15. Dhakal S, Jaafar F, Zavarsky P (2019) Private blockchain network for IoT device firmware integrity verification and update. In 2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE), IEEE, pp 164–170). https://doi.org/10.1109/hase.2019.00033
16. Martin KM (2017) Everyday cryptography: fundamental principles and applications, 2nd edn edn. Oxford University Press. https://www.ebook.de/de/product/28924485/keith_m_martin_everyday_cryptography.html
17. Rath S, Nguyen LD, Sahoo S et al (2023) Self-healing secure blockchain framework in microgrids. IEEE Trans On Smart Grid 14(6):4729–4740. https://doi.org/10.1109/tsg.2023.3253723
18. Rauchs M, Glidden A, Gordon B et al (2018) Distributed ledger technology systems: a conceptual framework. In: Techreport, Cambridge Centre for alternative finance. University of Cambridge. https://doi.org/10.2139/ssrn.3230013
19. Yaqoob I, Salah K, Jayaraman R et al (2021) Blockchain for healthcare data management: opportunities, challenges, and future recommendations. Neural Comput And Appl 34(14):11475–11490. https://doi.org/10.1007/s00521-020-05519-w
20. Schnitzbauer M (2021) Smart contracts in healthcare. In: Glauner P, Plugmann P, Lerzynski G (eds) Digitalization in healthcare - implementing innovation and artificial intelligence, 1st edn. Springer Cham, pp 211–223. https://doi.org/10.1007/978-3-030-65896-0_19
21. Islam MA, Islam MA, Jacky MAH et al (2023) Distributed ledger technology based integrated healthcare solution for Bangladesh. IEEE Access 11:51527–51556. https://doi.org/10.1109/access.2023.3279724
22. Ray PP, Dash D, Salah K et al (2021) Blockchain for IoT-Based healthcare: background, consensus, platforms, and use cases. IEEE Syst J 15(1):85–94. https://doi.org/10.1109/jsyst.2020.2963840
23. Yang X, Yue C, Zhang W et al (2024) SecuDB: an In-enclave privacy-preserving and tamper-resistant relational database. Proc VLDB Endow 17(12):3906–3919. https://doi.org/10.14778/3685800.3685815
24. Thantharate P, Thantharate A (2023) ZeroTrustBlock: enhancing security, privacy, and interoperability of sensitive data through ZeroTrust permissioned blockchain. Big Data and Cognit Comput 7(4):165. https://doi.org/10.3390/bdcc7040165
25. Akbulut S, Semantha FH, Azam S et al (2023) Designing a private and secure personal health records access management system: a solution based on IOTA distributed ledger technology. Sensors 23(11):5174. https://doi.org/10.3390/s23115174
26. Moya F, Quesada FJ, Martínez L et al (2023) Phonendo: a platform for publishing wearable data on distributed ledger technologies. Wireless Networks. https://doi.org/10.1007/s11276-023-03458-7
27. Jabbar R, Fetais N, Krichen M et al (2020) Blockchain technology for healthcare: enhancing shared electronic health record interoperability and integrity. In 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), IEEE, pp 310–317). https://doi.org/10.1109/iciot48696.2020.9089570
28. Namasudra S, Sharma P, Crespo RG et al (2023) Blockchain-based medical certificate generation and verification for IoT-Based healthcare systems. IEEE Consum Electron Mag 12(2):83–93. https://doi.org/10.1109/mce.2021.3140048
29. Peffers K, Tuunanen T, Rothenberger MA et al (2007) A design Science research methodology for information systems research. J Manag Inf Syst 24(3):45–77. https://doi.org/10.2753/mis0742-1222240302
30. Venable J, Pries-Heje J, Baskerville R (2016) FEDS: a framework for evaluation in design Science research. Eur J Inf Syst 25(1):77–89. https://doi.org/10.1057/ejis.2014.36
31. Herselman M, Botha A (2015) Evaluating an artifact in design Science research. In Proceedings of the 2015 Annual Research Conference on South African Institute of Computer Scientists and Information Technologists. ACM, SAICSIT 15, pp 1–10). https://doi.org/10.1145/2815782.2815806
32. Peffers K, Rothenberger M, Tuunanen T et al (2012) Design Science research evaluation. In: Chap design science research in information systems. Advances in theory and practice. Springer, Berlin Heidelberg, pp 398–410. https://doi.org/10.1007/978-3-642-29863-9_29
33. Yang X, Zhang R, Yue C et al (2023) VeDB: a software and hardware enabled trusted relational database. Proc ACM on Manag Of Data 1(2):1–27. https://doi.org/10.1145/3589774
34. Sim SH, Jeong YS (2021) Multi-blockchain-based IoT data processing techniques to ensure the integrity of IoT data in AIoT edge computing environments. Sensors 21(10):3515. https://doi.org/10.3390/s21103515
35. Yang X, Wang S, Li F et al (2022) Ubiquitous verification in centralized ledger database. In 2022 IEEE 38th International Conference on Data Engineering (ICDE), IEEE, pp 1808–1821). https://doi.org/10.1109/icde53745.2022.00181
36. go ethereum (2023) Clique signing. https://geth.ethereum.org/docs/tools/clef/clique-signing

Publisher's Note