electronics

Article

Blockchain-Based Security Configuration Management for ICT Systems

Dimitrios Chatziamanetoglou * and Konstantinos Rantos

Department of Computer Science, International Hellenic University, 65404 Kavala, Greece; krantos@cs.ihu.gr
* Correspondence: diehatz@cs.ihu.gr

Abstract: The world has become increasingly dependent on large-scale and distributed information and communication technology (ICT) infrastructures and systems in sectors such as energy, transport, banking, healthcare, water supply, and digital services, while their protection is considered of paramount importance and has already drawn remarkable attention from governments and key industry players. Establishing common approaches by leveraging existing frameworks and cyber security practices for improving the security postures of those systems is one of the major objectives for ensuring an adequate level of protection and avoiding the detrimental effects of disruptions on society and citizens. Configuration management (CM) is one of those common practices for establishing and maintaining the integrity and consistency of a system and its elements with regard to the function, performance, and status of technical and physical attributes, and it contributes to a desirable security posture throughout the lifecycle of a system. This study addresses the importance of CM, and while considering the corresponding frameworks, standards, and best practices, it proposes a permissioned blockchain-based approach, that inherits the benefits of the blockchain technology and ensures the integrity of the systems' configuration across the complete lifecycle management of its products and services as an underlying model for mapping and integrating CM functions. Furthermore, this study briefly presents the benefits and challenges of the application of permissioned blockchain models and proposes a smart-contract-based role-based access control mechanism, in addition to presenting an operating concept based on brief but real-life lifecycle requirements of organizational configuration management.

Keywords: configuration management; change management; blockchain; critical infrastructure; distributed and large-scale ICT

check for updates

Citation: Chatziamanetoglou, D.; Rantos, K. Blockchain-Based Security Configuration Management for ICT Systems. *Electronics* **2023**, *12*, 1879. https://doi.org/10.3390/electronics12081879

Academic Editor: Hamed Taherdoost

Received: 21 March 2023
Revised: 10 April 2023
Accepted: 13 April 2023
Published: 16 April 2023

1. Introduction

The cyber landscape is under constant changes as malicious actors exploit known cyber security gaps and discover new ones. Over recent years, cyber attacks on critical ICT systems have increased [1], and they have become more sophisticated and effective than before. The consequences of cyber attacks on a large-scale and distributed ICT infrastructure for financial, political, or military gain could include service degradation or disruption, environmental damage, financial loss and/or human injuries, or threats to human lives on a large scale, thus causing serious problems.

A fundamental understanding of how malware actors target critical systems and infrastructures can help organizations and key stakeholders comprehend how to conduct cyber security defense operations, respond to incidents, embed security in systems' design, understand risks and business impacts, implement strategic, operational, and tactical changes, and protect themselves from possible harm.

Collaboration between states and public or private entities facilitates the development of policies, frameworks, and guidelines for raising awareness, establishing best practices, leveraging and combining strengths, developing skills, identifying gaps, and increasing preparedness. These interactions also help teams work together more efficiently to deter, detect, and apply effective responses to cyber attacks, especially across distributed ICT infrastructure components, which, in many cases, deal with geographically dispersed legacy systems.

As cyber threats continue to evolve, organizations still have limited resources for minimizing their attack surfaces and improving their security postures. Security has become a risk-based activity, where the operational and economic costs define the appetite and tolerance thresholds of risk, and they are fully integrated to balance the needs of an organization's mission and business processes against cyber threats. In today's digital world, resources are scoped and tailored to fulfill this balance, and the practice of a risk management approach is fundamental to cyber security programs.

Information technology (IT) and operational technology (OT) systems are under a constant state of change, which spans technological, physical, business, and even human elements. Such changes could be hardware or software changes, incoming personnel with different skills, changes in system architectures, changes in the supply chain, changes in physical and technical access control measures, and many more. A disciplined and structured approach to monitoring, controlling, and documenting such changes is essential for the support of the security of IT and OT systems [2]. Such activities fall under configuration management (CM), the absence of which can have a significant impact on the security and privacy postures of these systems.

According to the National Institute of Standards and Technology (NIST) [3], Configuration Management is an umbrella definition that covers a collection of activities that are focused on establishing and maintaining the integrity of information technology products and information systems through the control of processes for initializing, changing, and monitoring the configurations of those products and systems throughout the system development lifecycle. Change management—despite being frequently shown as a standalone function—is a specific functional element of configuration management that is responsible for managing changes during the lifespan of a system. Its interactions with the rest of the configuration management functions will be shown and described below.

The concepts, principles, and processes described in existing publications set the overarching high-level framework of security configuration management (SCM) [3–5] with the objective of managing and monitoring the configurations of information and operational systems to achieve adequate security and minimize organizational risks while supporting the desired business outcomes and services.

Furthermore, during the rapid development and wide application of distributed ICT systems, the interest in blockchain technology has dramatically increased and has made it a widely accepted solution due to the efficiency, applicability, and essentials of its features [6]. Blockchain technology can be used to cryptographically sign the "who", "what", "where", "when", and "why" for the status of and changes in all critical cyber assets throughout the chain of custody. Traceability, auditability, decentralization, immutability, transparency, and peer communications are features of distributed ledger technology that are enablers in supporting the management and security of information technology products and information systems more effectively [7].

Focusing on the principles of security configuration management, this study proposes a blockchain-based model as the foundation of configuration management and change control applications; this model inherits all of the benefits and advantages of the integrity, tamper-resistance, trust, and scalability of distributed ledgers. Placing an emphasis on the protection of distributed ICT infrastructure and its complex nature, this study addresses the requirements of CM by proposing a permissioned ledger as a closed ecosystem with a defined governance structure, private transactions, and strict authentication access in order to maintain the security requirements of the critical components of IT/OT infrastructures.

The aim of this study is two-fold. The first objective is to present a high-level overview of the existing frameworks, publications, handbooks, and guidelines that underpin the CM process in IT/OT systems, especially in the domain of large-scale and distributed systems. The second objective, which is of equal importance, is to map the fundamental processes, practices, and phases of CM and demonstrate how those can be applied in a blockchain model to support real-life technical requirements by benefiting from the advantages and features inherited from distributed ledger technology.

The rest of this paper is structured as follows. Section 2 presents the existing set of publications, frameworks, handbooks, and guidelines that define CM, as well as the existing background research related to the scope. Section 3 presents the motivations and challenges that triggered the proposal of a blockchain-based CM model, since the existing research was found to be limited. Section 4 briefly presents the CM process and the phases that it comprises, as well as related definitions. Section 5 presents the proposed blockchain-based model and a related mapping of the functions of CM. Section 6 describes the operating concept using a practical application of a complete lifecycle of the CM process, while Section 7 summarizes the paper's proposals and depicts our objectives for future work.

2. Background and Research Review

Configuration management is a strong requirement in many cyber security frameworks and standards. It is used to enable the functional and physical attributes of IT/OT platforms, products, and their environments to determine the appropriate security features and assurances that are used to measure a system configuration state. It is also used to control modifications to hardware, firmware, software, and documentation in order to ensure that ICT systems are protected against unauthorized modifications prior to, during, and after system implementation by establishing baseline controls via policies, practices, and procedures.

The following publications are used to define the set of requirements, starting from risk management, ending with the implementation of configuration management, and spanning across various governmental and institutional entities:

1. ANSI/EIA-649C, Configuration Management Standard
2. Cloud Security Alliance (CSA) Cloud Controls Matrix (CCM)
3. EIA-836B, Standard for Configuration Management Data Exchange and Interoperability
4. ENISA Risk Management/Risk Assessment Framework
5. European Council Directive 2008/114/EC
6. IEC62443-Security for industrial automation and control systems-Parts 2-4, 4-1 and 4-2
7. ISO/IEC 27001, Information Security Management
8. ISO/IEC 20000:2018, IT Service Management System Requirements [8]
9. NIST Special Publication (SP) 800-37 Revision 2: Risk Management Framework for Information Systems and Organizations [2]
10. NIST SP 800-53 Revision 5: Security and Privacy Controls for Information Systems and Organizations [4]
11. NIST SP 800-82 Revision 2: Guide to Industrial Control Systems (ICS) Security [5]
12. NIST SP 800-128: Guide for Security-Focused Configuration Management of Information Systems [3]
13. NIST SP 800-160: Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems [9]
14. NIST Cybersecurity Framework: Framework for Improving Critical Infrastructure Cybersecurity [10]
15. US DoD Protected Critical Infrastructure Information (PCII) Program

Alongside the above-mentioned frameworks, there are several other libraries of frameworks, standards, and best practices [11] that are focused on IT service management activities for service providers, enterprises, and military organizations. These publications play a significant role in the overall definition of CM and the set of requirements, processes, procedures, and business outcomes. These can be listed as follows:

1. ENISA Guidelines on Security Measures under the European Electronic Communications Code (EECC)

2. COBIT 2019, Management and Governance of Enterprise IT [12]
3. ISO 10007:2017, Quality management—Guidelines for configuration management
4. ITIL v4, IT Infrastructure Library
5. Mil-HDBK-61B, Configuration Management guidance [13]
6. TM Forum Business Process Framework (eTOM)

In the scope of the aforementioned publications that set the high-level frameworks and procedural guidelines, there are plenty of works found in the literature that address implementation proposals that follow the requirements of configuration management, but the research activity on blockchain-based solutions for supporting CM implementations is very limited.

Kinkelin et al. [14] proposed an abstract Byzantine fault-tolerant (BFT) configuration management system (CMS) based on the Hyperledger Fabric environment with the objective of managing configuration requests in an operational environment. Their proposal acts as an intermediate authority between administrators and managed devices, and it is able to conduct multi-party authorization for critical configurations to prevent individual malicious administrators from performing undesired actions; changes are applied only after a configuration has been validated and authorized by multiple experts. One of the drawbacks of the proposed system was found to be the potentially low number of validators, which would weaken the protection of this CMS. Kostal et al. [15] proposed a private blockchain approach for storing and loading configurations of Internet of Things (IoT) devices to manage and monitor network devices, and they highlighted the tamper-proof functionalities of the proposed method, which was also supported by off-chain databases. Furthermore, authorized network administrators used digital certificates to authenticate themselves, while they could modify the configuration of devices if they were authorized to do so for a given device or group/domain of devices. Alvarenga et al. [16] proposed a blockchain-based architecture for secure management, configuration, and migration of virtualized network functions, which ensured the immutability, non-repudiation, and auditability of the configuration update history and the anonymity of tenants and configuration information while guaranteeing the secure update and migration of configurations at the core of the network and being resilient to collusion attacks from up to one-third of the blockchain modules. Mylrea et al. [17] examined how blockchain technology can enable critical infrastructure protection compliance and aid in the security of software supply chains, patch management, and configuration management through an immutable cryptographically signed distributed ledger that enabled improved data security, provenance, and auditability while describing the challenges in applying blockchain technology due to the lack of existing policies and governance. Han [18] proposed a very abstract blockchain configuration management system that could protect the copyrights of software development projects and systems' configurations. Samaniego et al. [19] proposed a limited-scope virtualization of IoT components with the objective of supporting configuration management and provision across an IoT network, and they achieved efficiency in terms of latency and bandwidth. The solution was based on a permissioned blockchain with encrypted blocks for additional security.

The aforementioned research touched some elements of the overarching CM process, but did so in isolation—without a contextualized approach—and, most importantly, without identifying which function of CM they were addressing and how the proposed work might interact with the rest of the functions of CM. The present study covers this gap by addressing the challenges and opportunities of an efficient implementation of a CM process and presents an end-to-end blockchain-based model that enables all of the different functions of CM. Furthermore, it demonstrates how the benefits of blockchain technology can be leveraged in the CM lifecycle in alignment with existing frameworks, policies, recommendations, guidelines, and best practices.

3. Motivation and Challenges

Industries, enterprises, and governmental organizations use a plethora of tools that cover areas of the requirements of configuration management [20]. These tools have a certain overlapping coverage area. This is the reason for why more than one tool is required to cover the whole scope. Furthermore, there is no indication that these tools currently utilize blockchain technology in their production environments, especially for configuration management purposes.

We identified that even though there is a very mature framework addressing the requirements of that stem from the frameworks, policies, guidelines, and best practices of CM that were presented before, there is not a mature implementation showing how these requirements can be applied in a blockchain infrastructure by utilizing the value and benefits of this technology. In addition, the majority of the existing research covered CM topics in an abstract manner and did not address a holistic proposal or solution or how this may interact with the rest of the functions of CM.

It was found that the need for the development and application of blockchain technology to underpin the requirements of configuration management is evolving, and it is expected to expand and grow; when used, this technology will bring additional value and boost the business outcomes and overall security of large-scale and distributed ICT systems [17].

4. Configuration Management

4.1. Definitions and Value of Configuration Management

Information systems are discrete sets of information resources that are organized for the collection, processing, maintenance, use, sharing, dissemination, or disposition of information, and they are composed of several components, the configuration of which has a direct impact on the security posture and the operational functionality of the system [4]. A configuration item (CI) is a single component or set of components of an information system that is subject to configuration management and is considered as a single entity throughout the practice of configuration management [3,13]. Each CI should be properly identified, labeled, and tracked during its lifecycle, its interactions, and its contributions to an overall system's function. A CI could be a network element, a server, an application, a documentation, a security compliance checklist, a service, or even the information system as a whole, which when defined correctly, provides an organization with the means to apply the desired lifecycle management for security and operational requirements. Each CI has a baseline configuration.

A baseline configuration is a set of specifications for a CI or a set of CIs within a system that have been formally presented, reviewed, and approved at a given point in time. The baseline configuration can only be changed through a change control procedure triggered by a change request. The baseline configuration is used as a basis for future builds, releases, or changes and evolves over time depending on the stage and progress of the system development lifecycle (SDLC) requirements, such as development, testing, production and retirement. Early in the SDLC, when a system is being initiated and acquired, the baseline may be a set of functional requirements. As the system is developed and implemented, the baseline may expand to include additional configuration elements to fulfill the end-state objectives of production. When a new baseline configuration is established, all of the changes from the last baseline are approved. Older versions of approved baseline configurations are maintained and made available for review or rollback as required. There are also different types of baselines, such as functional baselines, product baselines, service baselines, etc., the differences of which, will be left out of this study.

CI records contain all information related to each CI, including the baseline configuration, unique identifiers, description, version, location, type, relationships with other CIs, status, etc. Practically, this information may include as many attributes/properties that can uniquely and unambiguously describe a CI, depending on the data model used by the system.

The way the configuration of the CIs is implemented, maintained, and managed requires a disciplined approach to providing adequate security and functionality and fulfilling the organizational outcomes and objectives. Changes in the CIs are often required to fulfill business requirements, adopt IT architectural changes, react to incidents and known IT problems, and be up to date on security needs. These changes can heavily and negatively impact the previously established security posture and/or operational requirements. This is the reason why a controlled, documented, and effective configuration management process is critical for the maintenance and improvement of the security posture, functional outcomes, and business objectives of an organization.

Security-focused configuration management (SecCM) is the management and control of secure configurations for an information system to enable security and facilitate the management of risk [3,21], which is still a subset of the overall configuration management process. The security-focused configuration management process is vital in maintaining a secure state during organizational operational management, ensuring that risks are properly assessed and changes are authorized to proceed, and managing a reliable and updated change record that is always available for auditing and accounting purposes.

A configuration management database (CMDB) is used to store configuration records throughout their lifecycle and maintain the relationships among them. It also helps an organization understand the relationships between the components of a system and track their configurations.

A configuration management system (CMS) comprises a minimum of one CMDB, while multiple CMDBs can be used by an organization to store and manage CI records from different domains. In addition, a CMS contains information related to incidents, problems, and known errors, which are functions that fall outside the scope of this study. The immutability of a CMS is considered vital, as it constitutes the overall IT baseline of the whole organization.

In practical terms, a CMS is part of a larger system called a service knowledge management system (SKMS). The SKMS consolidates and analyzes configuration item (CI) records to facilitate the design, development, delivery, operation, and improvement of services. The SKMS is based on a number of best practices and industry standards [8,12], and it fulfills organizational service management, security, and business intelligence objectives and requirements.

Finally, the SKMS also includes a definitive media library (DML), which is a secure repository in which an organization stores definitive and authorized versions of software, media, and data. When there is a requirement for the deployment of a new release, only available releases existing in the definitive media library can be used to build the new release. The entries in the DML can be considered as separate CIs, the records of which can be stored in the aforementioned CMDBs.

4.2. Configuration Management Functions

In general terms, the aforementioned configuration management (CM) standards, handbooks, and publications define the configuration management process with five key functions/disciplines [3,9,13] (shown in Figure 1):

- Management and planning;
- Configuration identification;
- Configuration control or change management (ChM);
- Configuration status accounting;
- Configuration verification, evaluation, and auditing.

Management and planning deal with defining the strategic aspects that are required to be in place in order to underpin the next stages. These aspects include the definition of roles, responsibilities, tools, interfacing processes, organizational criteria, and priorities, the definition of configuration items (CIs) and baselines, security considerations, conditions, and constraints, the development of documents such as directives, operational procedures, and guidelines, and the establishment of organizational boards that will enforce and oversee the configuration management process.

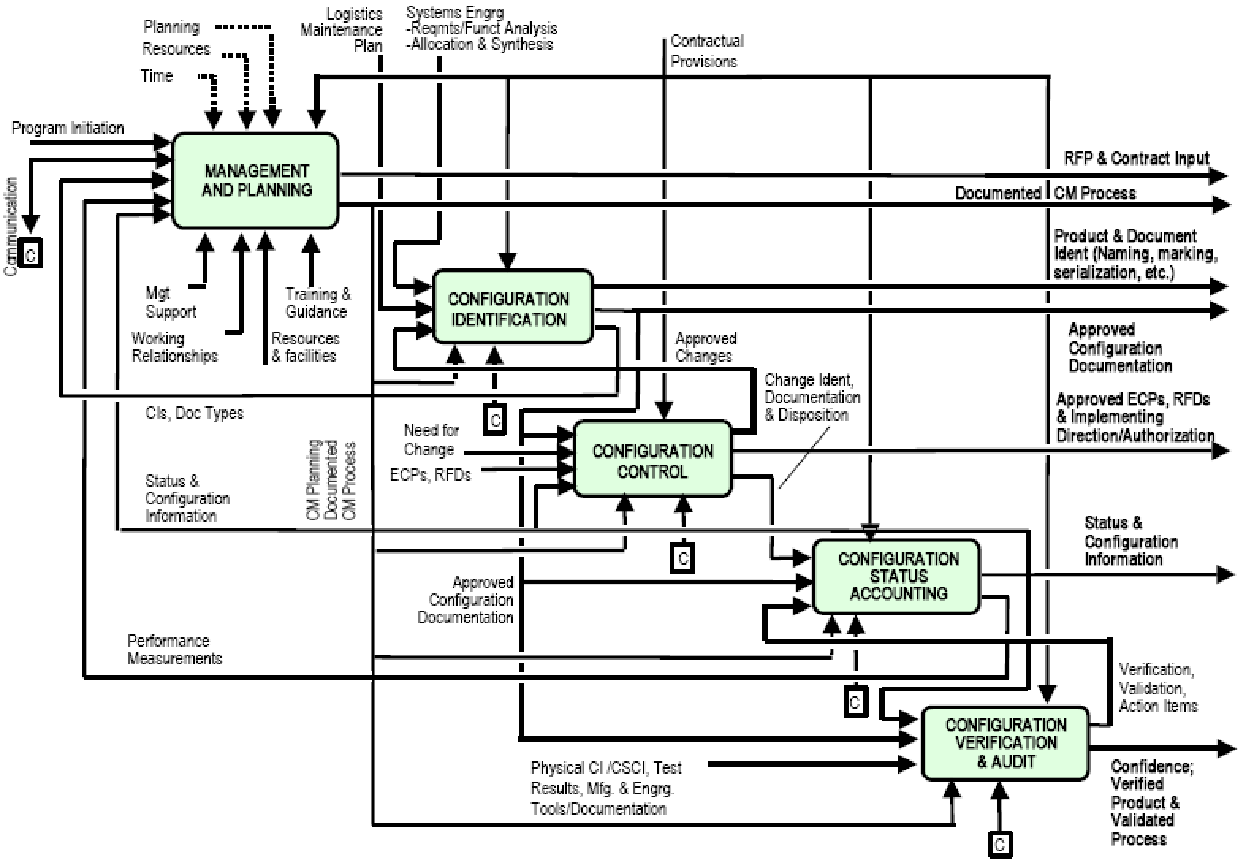
Planning
Resources
Time
Logistics Maintenance Plan
Systems Engrg -Reqmts/Funct Analysis -Allocation & Synthesis
Contractual Provisions
Program Initiation
MANAGEMENT AND PLANNING
RFP & Contract Input
Documented CM Process
Communication
C
Mgt Support
Working Relationships
Training & Guidance
Resources & facilities
CONFIGURATION IDENTIFICATION
Product & Document Ident (Naming, marking, serialization, etc.)
Approved Changes
Approved Configuration Documentation
Change Ident, Documentation & Disposition
Approved ECPs, RFDs & Implementing Direction/Authorization
CIs, Doc Types
Need for Change
ECPs, RFDs
CONFIGURATION CONTROL
Status & Configuration Information
CM Planning Documented CM Process
Status & Configuration Information
CONFIGURATION STATUS ACCOUNTING
Approved Configuration Documentation
Performance Measurements
Verification, Validation, Action Items
CONFIGURATION VERIFICATION & AUDIT
Physical CI /CSCI, Test Results, Mfg. & Engrg. Tools/Documentation
Confidence; Verified Product & Validated Process

Figure 1. Configuration management functions (Mil-HDBK-61B [13]).

Configuration identification addresses the aspects of the proper identification, baseline, labeling, hierarchy, structure, and dependencies of CIs, which are subject to formal reviews and configuration audits. Via this function, all CIs should be uniquely and unambiguously identified, while the status of the configuration should be traceable for every past, present, or future (planned) configuration state.

Configuration control or change management (ChM) enforces the required security and control measures to maintain the secure, approved baseline of the system, minimize unauthorized and/or undocumented changes, further manage, document, and coordinate among all stakeholders, evaluate the quality, benefits, and costs, and assess the risk of all requested changes during the lifecycle management of all CIs by applying a broad range of procedures and criteria in order to finally deny or approve all change requests via the organizational regulatory boards, which are normally called configuration control boards (CCBs) or change advisory boards (CABs).

The configuration status accounting function captures, stores, maintains, and processes configuration management status information for CIs with respect to their baselines, approved changes, and releases in order to preserve, protect, and ensure the integrity, confidentiality, and availability of the CIs by supporting the planning and decision making for certification, accreditation, authorization, and reporting activities. By preserving current, accurate, and retrievable information on the status and configuration of CIs, an organization can ensure the effective and efficient lifecycle management of its systems, starting from the design phase, testing, and production and ending with full retirement.

The configuration verification, evaluation, and auditing function includes all necessary technical and non-technical capabilities for addressing security and operational concerns. These capabilities include security compliance checks, configuration gap analysis, and the difference between the “as-certified” and “as-maintained” status of CIs. Via these audit control mechanisms, an organization is able to assess whether a system conforms to the security aspects of the operational requirements against the defined baselines and is capable of assessing security-relevant discrepancies, variances, and deficiencies.

5. Proposed Model

This study consolidates the principles of configuration management and proposes a blockchain-based design that inherits all of the benefits and advantages of the integrity and immunity of distributed ledgers.

As mentioned before, the configuration management system is considered the overall baseline of a whole organization; thus, its confidentiality, integrity, and availability (CIA triad) are meant to be vital for the security posture and fulfillment of the operational objectives, especially when the organization underpins distributed ICT systems. The high-level components of the functions of CM and their proposed interrelationships with other functions and processes are shown in Figure 2, where the elements in green are enabled by blockchain technology.

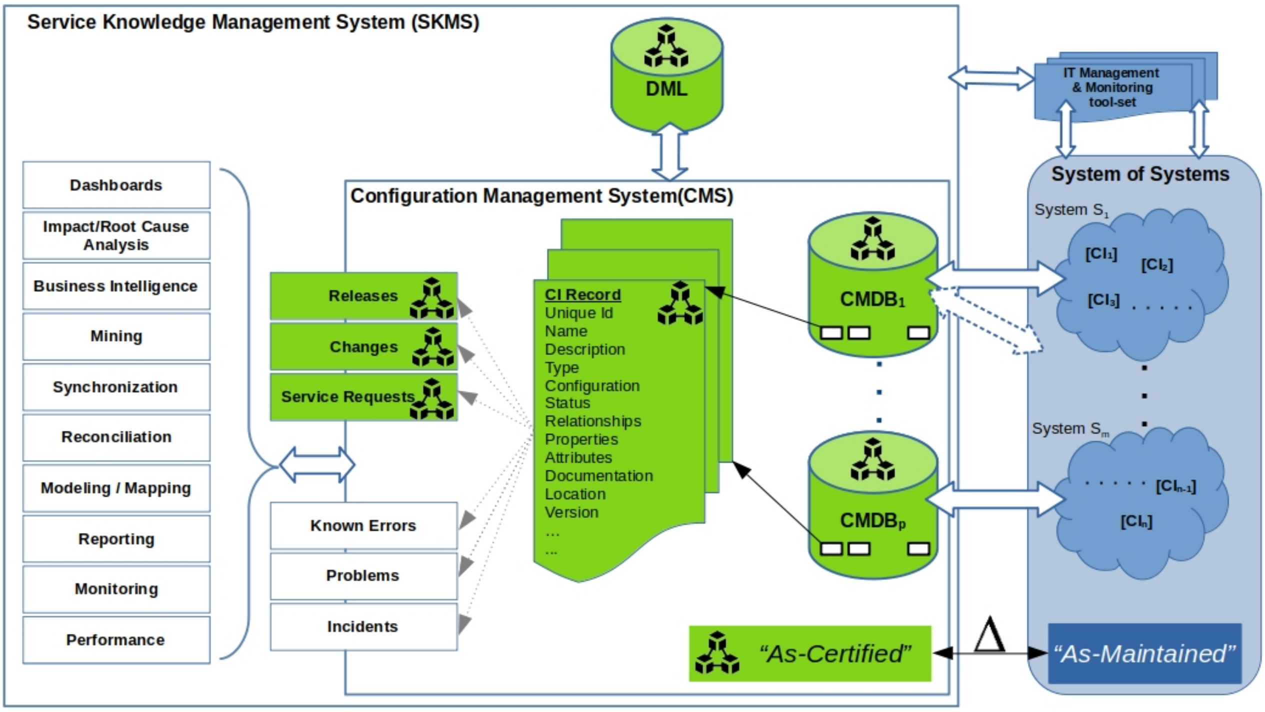

Figure 2. High-level components of CM.

A blockchain solution can be classified as public/permissionless or private/permissioned. Each one has differences in the nature of stakeholders' membership and authorization, which is required for participating in a blockchain network. In a public blockchain, anyone can register and interact without permission, while a permissioned network requires additional levels of access restriction policies, which narrow down the access, privileges, and rights of the participants.

Considering the increased security requirements and the nature of large-scale and distributed ICT systems, we propose the implementation of a permissioned distributed ledger with a well-defined governance structure, private transactions, and strict authentication for access in order to preserve and ensure the security requirements of critical IT/OT components. A permissioned ledger is proposed in order to allow only selected and verified participants of the organization to interact with the system by applying a role-based access control mechanism for giving specific privileges and access rights according to the roles and responsibilities in the CM process. Furthermore, the proposed blockchain-based system acts as a secure repository to ensure that all of the outputs of the configuration management process have not been tampered with.

This section will present the high-level architecture of the model by depicting all of the interfacing components and elements of the CM process, as well as the dependencies of the CIs of an IT system, while following an agnostic data model approach.

5.1. "As-Certified" vs. "As-Maintained"

Figure 2 depicts a helicopter view of the elements underpinning the concept of configuration management. The left part depicts the SKMS, which was defined earlier and represents the holistic configuration baseline of an organization, while the right part depicts the real-life status of the deployed IT/OT elements that serve the operational objectives of the enterprise.

An organization should maintain the status of its CIs, including the information on the stages of the whole lifecycle of those CIs, such as their design, planning, and status of being delivered/implemented, by utilizing various tools. These tools provide not only the capability of maintaining the baseline (golden data) of the organization, but also provide various levels of functionalities in order to model, map, reconcile, synchronize, analyze, report, and present the status of those CI records in a combined and controlled manner via either detailed or even executive-level views/dashboards. In our proposal, the core elements of the SKMS that hold the critical information of the organizational baseline are stored on a blockchain (green color) and benefit from its inherited advantages, while the rest of the depicted functions can be supported by conventional and existing methodologies/toolsets.

As mentioned before, the CIs of an organization are considered the building blocks of that organization's ICT infrastructure, and they consist of hardware, software, documentation, and many other components, which are either tangible or not. Proper management of CIs is critical for ensuring the availability, reliability, and security of ICT systems and services. Product lifecycle management (PLM) is a comprehensive approach that enables organizations to manage CIs throughout their lifecycle. Figure 3 depicts the lifecycle of a product from "cradle to grave", starting from the requirements and development of operational capabilities and ending with retirement and decommission [22]. Among all of the intermediate stages of a product lifecycle, in this study, we chose the "As-Certified" and the "As-Maintained" stages, which represent the "latest approved" baseline and the "As-Is" status of a CI, respectively, while in [13], these statuses were reflected as the "as-designed configuration" and "as-built configuration", respectively. These concepts can also be conceptually borrowed from product information modeling, building information modeling, and product lifecycle management [23–25], in conjunction with the digital thread and digital twin approaches [26–28].

On one hand, the set of information that constitutes the "As-Certified" status of the CIs of an ICT system of systems (SoS) includes the configuration status of the final approved design (latest approved baseline), including all changes authorized so far during the lifecycle of each CI. On the other hand, the real-life or actual configuration status of the deployed CIs constitutes the "As-Maintained" status of the SoS.

There are many ways to obtain the "As-Maintained" (or As-Is) status of the CIs, either by technical means (SNMP polling, network telemetry subscription, compliance checks, etc.) by utilizing specific IT management system tools, by physical means (inventory checks, site surveys, etc.), or even a combination of the above.

In theory, the "As-Certified" and "As-Maintained" status of each CI must be exactly the same, which means that everything that is implemented under the operational SoS should strictly follow the designed, planned, and authorized requirements. In reality, the situation is slightly different, as there is a mismatch between statuses, and organizations strive to keep this gap as short as possible. There are many reasons for having differences between the As-Certified and As-Maintained statuses, and some examples are provided below:

- Unauthorized changes;

- Emergency change implementations (e.g., responding to a major incident) with no post-documentation/CMS integration;
- Lack of change control/change management processes;
- Deviation from processes and guidelines;
- Malicious activities by insiders/outsiders;
- Malware that impacts the CI configuration;
- Wrongly or partially implemented changes;
- Failed changes without roll-back;
- Deficiencies during hand-over or take-over activities from project deliveries.

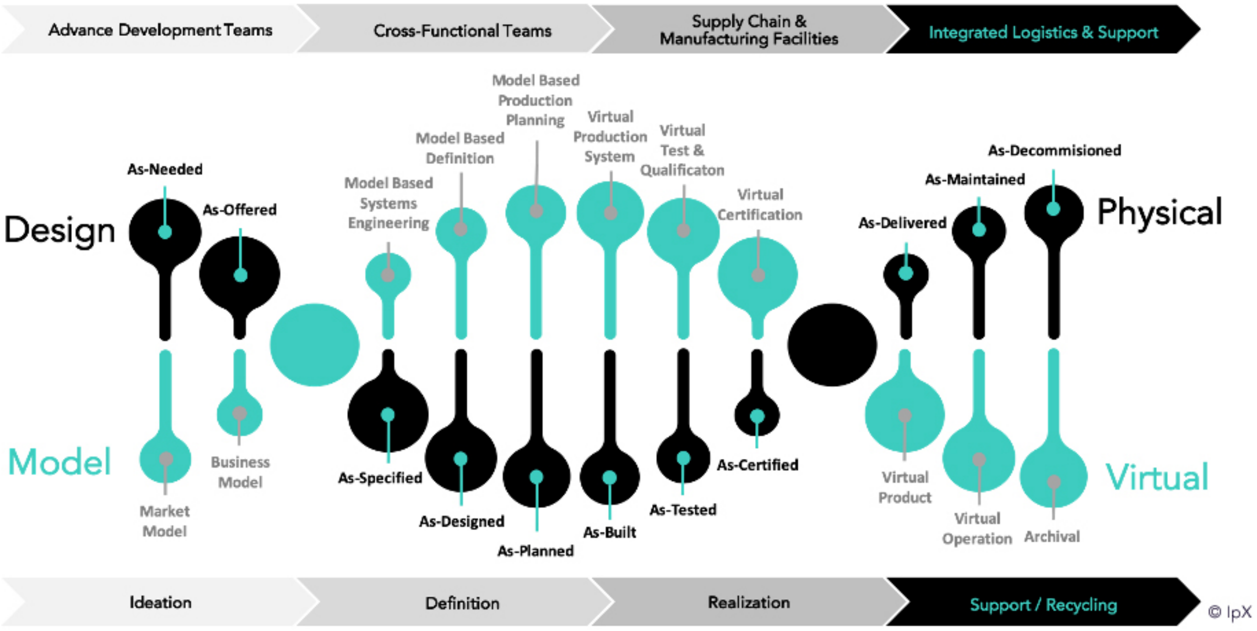

Figure 3. Product lifecycle [22].

5.2. Description, Benefits, and Limitations of the Blockchain CM Model

The type of blockchain proposed in this study is a permissioned ledger in order to provide increased security, transparency, and efficiency and allow for a more controlled and regulated environment for participants, thus enabling higher levels of trust and collaboration. Permissioned blockchains are increasingly used in the industry and their applicability is constantly benchmarked and evaluated. Depending on the application domain, blockchain technology has benefits and drawbacks compared to traditional databases and approaches [29–31], but there is continuous effort being made to identify gaps and challenges in terms of performance (execution time, latency, and throughput), scalability, and applicability [32–34] in order to meet the evolving technological requirements by indicating suitable strategies that can be deployed in blockchain systems [35–40].

The present model takes advantage of the existing features and benefits of distributed ledger technology and uses them as a vehicle for a new application area in the domain of configuration management. In essence, the main benefits of using blockchain technology—and, in particular, a permissioned blockchain—in support of security configuration management are as follows:

1. Tamper-proof CI records that enable integrity and constitute an immutable inventory baseline of an entire organization.
2. Historical records that enable traceability, verification, validation, auditability, restoration, and data recovery.
3. Transparency, which underpins the configuration status and accountability of actions and changes.

4. A shared ledger that enables the secure distribution of baseline configurations in the community.
5. Increased privacy and confidentiality via access control (AC) mechanisms by using smart contracts.

All of the above can further underpin real-time, near-real-time, or ad hoc/planned security compliance checks in order to detect malware or unauthorized changes/activities.

In addition, access control is a critical aspect of permissioned blockchain systems, which require the robust management of privileges and rights to ensure the integrity of the ledger and the security of the information stored on it. The application of CM requires a clear definition of specific roles and responsibilities for their functions to be performed inside an organization, such as in planning, submitting, and approving changes in CIs, as well as releasing and deploying changes and accessing data in the ledger. Hence, role-based access control (RBAC) is deemed necessary as a control mechanism to enable the implementation of a more scalable and manageable access control system by grouping users with similar roles together and then assigning them the appropriate permissions. Smart contracts can be used to enforce these access control policies [41] to provide an embedded, robust, flexible, and transparent mechanism for managing such permissions [42–46]. On the other hand, the effective implementation of RBAC requires careful planning, documentation, and continuous monitoring to ensure that access privileges remain appropriate and up to date.

The main stages of the configuration management process are shown in Figure 4, along with the interacting elements of the proposed blockchain-based model. The CM process is mainly a feedback-based cyclic model and conceptually starts with configuration management planning, where the governance, strategy, roles, responsibilities, and modeling are given a place on the organizational level, followed by the identification of the configuration items and their structure, relationships, and dependencies inside the system. These first two stages of the CM process do not interact with the blockchain model that we propose but are critical enablers for the following actions.

As mentioned before, the configuration control stage deals with the control of the changes in all configuration items of a system. In our proposal, it is clearly shown that the configuration management system (CMS) is completely based on the blockchain (highlighted in green), which leads to the conclusion that the overall baseline of all CIs of an organization benefits from the features of the immutable ledger. This design further leads to the outcome that the “As-Certified” status of the organization’s critical elements or assets is stored in such a way that it grants full control and complete tracking of all lifecycle changes; it is also leveraged by the blockchain features that underpin the configuration status accounting stage of the CM process. In parallel, the “As-Maintained” status of an organization’s ICT system is retrieved and made available via the applied technical monitoring and management solutions to show the current status of the CIs.

The combination of the above can lead to the detailed identification of the difference (delta) between the two statuses (“As-Certified” vs “As-Maintained”) at a level of granularity that fulfills the expectations of an organization’s auditing requirements. This delta is one of the fundamental benefits of the overall configuration management process; it provides a clear picture of the “As-Should-Be” vs. “As-Is” status of the CIs in the configuration, verification, and auditing stage of the CM process.

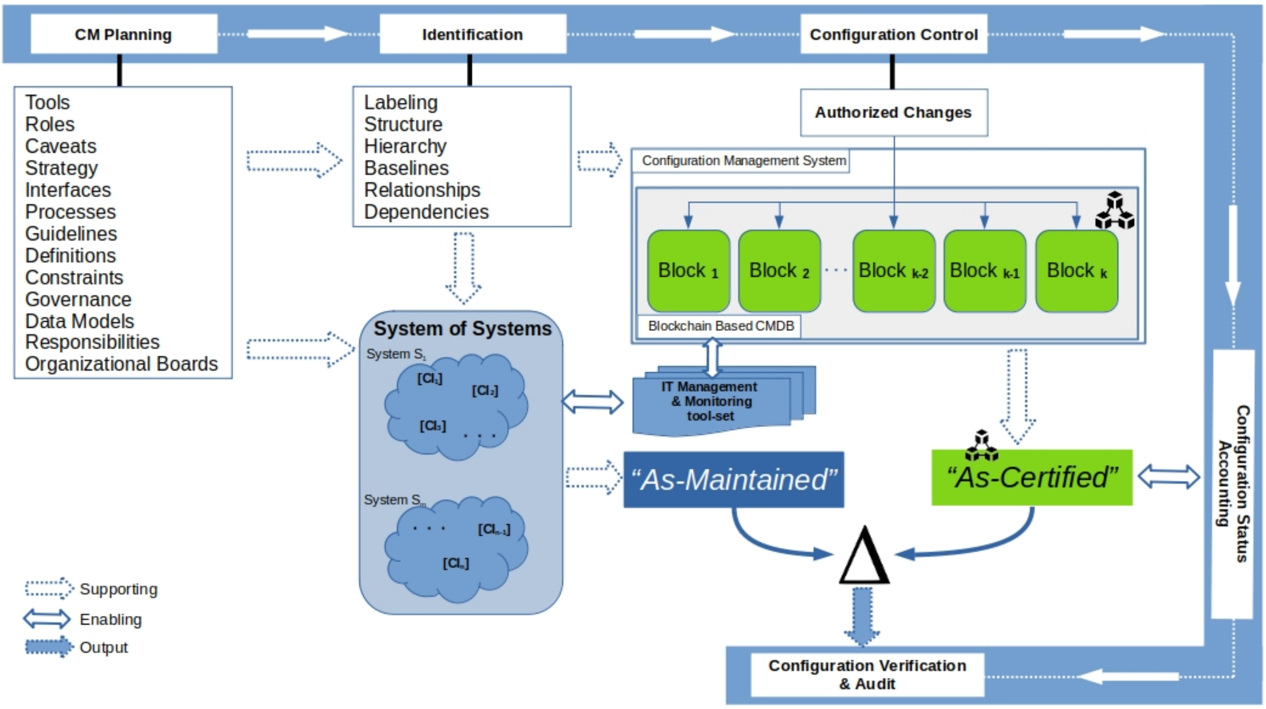

Figure 4. Configuration management process.

The identification of such differences can be applied in quantitative and qualitative analyses and is considered critical because an organization can assess the level of deviation of its configuration baselines, which can lead to more effective performance and more efficient compliance checks.

These compliance checks are not only applicable from the perspective of security, but also from many other perspectives that are linked together by the overall organizational requirements, such as asset management, release and deployment management, vulnerability assessments, financial management, and overall risk management activities.

6. Operating Concept

This section presents an operating concept of the proposed model, and a use case in a configuration management scenario is presented. For this use case, four entities with high-level roles and responsibilities are represented as follows:

- The change requestor's role is to plan, find the resources for, develop, test, and support the proposed changes in one or more configuration items of the system, as well as to present them to the configuration control board;
- The configuration control board (CCB) has the role of establishing and chartering a group of qualified people with responsibility for the process of assessing, controlling, approving, and recording changes throughout the development and operational lifecycle of the system;
- The implementation team has the role of releasing and deploying the approved changes;
- The security auditor has the duties of performing security compliance checks, configuration gap analyses, and post-implementation checks and comparing differences between the "As-Certified" and "As-Maintained" statuses of the system's CIs.

Figure 5 shows a flowchart of the actions performed by the defined actors when executing the roles mentioned above. More specifically, the change requestor defines the scope of the change, justifies its necessity, identifies and assesses the impacts on other configuration items, develops the test and coordination plan, and includes all of the other necessary documentation, such as the deployment, support, roll-back plan, etc., as required by the internal procedures of the organization. The change proposal is submitted as a technical package to the configuration control board (CCB) for the change evaluation by using existing IT service management tools.

The CCB reviews the change proposal, assesses the risk to the organization, and performs compliance checks as set by the organization, and it approves the change, rejects it, or escalates it to a higher-level CCB. If the change is escalated to a higher-level CCB, the next-level CCB performs the review and assessment of the change against a wider audience and additional criteria with two output options: change approval and change rejection.

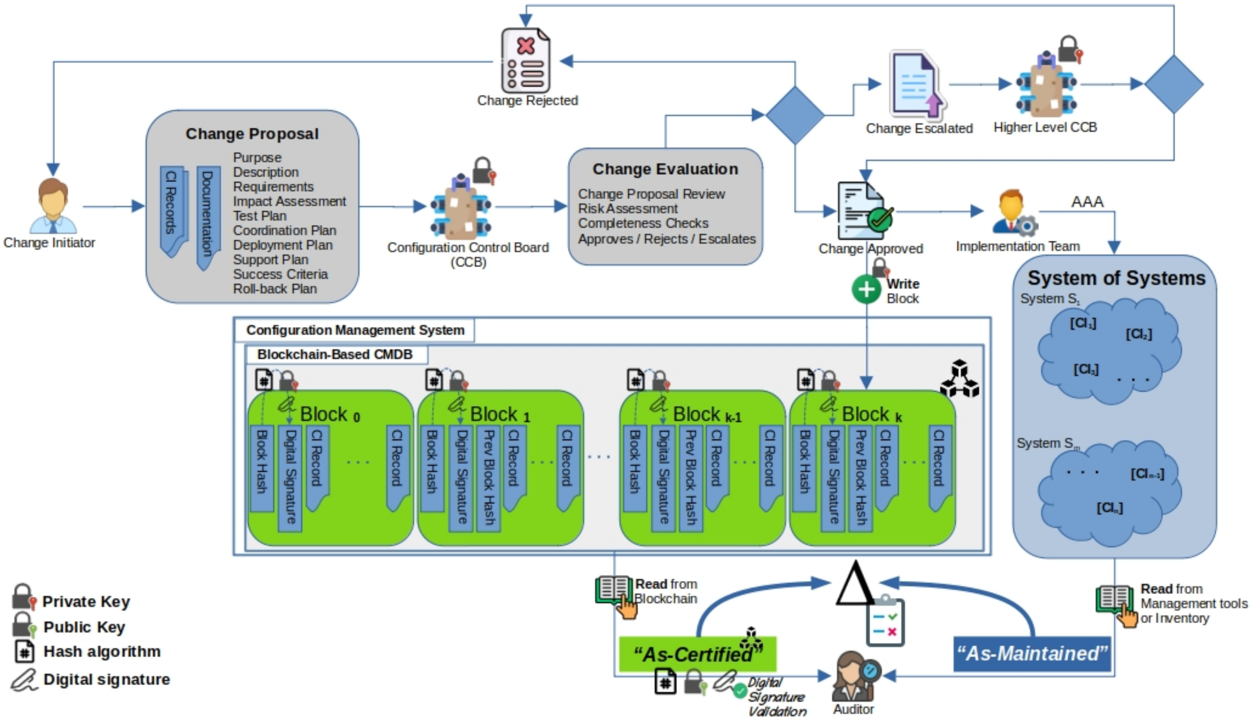

Figure 5. Flowchart of the operating concept.

If the change proposal is rejected for any reason, it is returned to the change requestor with full justification for further adjustments and further re-submission if deemed necessary. If the change proposal is approved, then the full details of the change are stored in the blockchain as a separate block, while, in the meantime, it is pipelined to the implementation team to streamline the change's implementation.

It is clear that any changes that are approved to be applied in the operational system can take place only after a change is approved by the configuration control board. Similarly, the same applies for adding blocks to the blockchain-based configuration management system (CMS). This implies that the "As-Certified" status of the system is under full control by using the features of the blockchain; moreover, a digital signature mechanism is applied for additional non-repudiation and accountability purposes [47]. On the other hand, the accountability and the non-repudiation in the operational ICT system are expected to be based on equivalent technical functions built on the system's management tools, such as AAA (authentication–authorization–accounting) mechanisms, which fall outside the scope of this study.

A digital signature is applied to the hash of the block that is planned to be added to the blockchain and further stored as separate data on the block itself. The digital signature ensures the identification of the change approval entity by utilizing the private keys of the involved CCBs that are authorized to approve changes. The use of a digital signature functionality requires an external public key infrastructure (PKI) to be in place in order to support the certificate's chain of trust. The establishment of a PKI is not covered in this study.

Block-0 represents the genesis block of the blockchain and contains the initial status and information of all configuration items in the system, thus representing the initial baseline; it is digitally signed by the highest-level CCB of the organization. The following blocks hold the information of the configuration items, which are the subjects of each change and are digitally signed by the authority that approved each change, thus progressively building the final/current approved ("As-Certified") baseline of the system.

As explained above, the output of the configuration, verification, and auditing process is based on the comparison between the "As-Certified" and "As-Maintained" statuses of the system being audited. The auditor verifies the digital signature of the blockchain blocks whose configuration items' records are within the scope of the audit, by using the public keys of the CCBs.

In essence, the proposed model underpins and enables the functions of the configuration management process, takes advantage of the existing benefits of blockchain technology, and ensures that all changes are securely recorded and added only after their approval and only by the authorized organizational elements. Furthermore, the immutable ledger can be used for configuration auditing purposes and for technical comparisons of the "As-Certified" and "As-Maintained" statuses of the system in order to identify unauthorized changes, security configuration gaps, lacking security compliance, wrong post-implementation changes, or potentially malware-impacted configuration items. All of the above are supported by a digital signature function that adds an additional security element of a non-repudiation feature to the ledger.

Finally, in support of the RBAC mechanism that was mentioned before, we present a brief code example of a smart contract in Listing 1 that implements just one simple feature. In this case, the smart contract owner can specify the authorized entities that can make changes in the ledger and add a configuration item into the ledger with the required information.

Listing 1. Smart Contract Code Example.

```
pragma solidity ^0.8.19;

contract ConfigurationItemRegistry {
    address public owner;
    mapping(address => bool) authorized;
    uint public authorizedCount;

    struct ConfigurationItem {
        uint uniqueId;
        uint configurationItemId;
        string description;
        string itemType;
        string configuration;
        uint timestamp;
        uint userId;
    }

    mapping(uint => ConfigurationItem) public configurationItems;

    constructor() {
        owner = msg.sender;
        authorized[msg.sender] = true;
        authorizedCount++;
    }

    modifier onlyOwner() {
        require(msg.sender == owner, "Only owners can perform this action");
        _;
    }

    modifier onlyAuthorized() {
        require(authorized[msg.sender], "Only authorized parties can perform this action");
        _;
    }

    function addAuthorized(address account) public onlyOwner {
        authorized[account] = true;
        authorizedCount++;
    }

    function removeAuthorized(address account) public onlyOwner {
        authorized[account] = false;
        authorizedCount--;
    }

    function addConfigurationItem(uint uniqueId, uint configurationItemId, string description, string itemType, string configuration, uint timestamp, uint userId) public onlyAuthorized {
        ConfigurationItem memory newItem = ConfigurationItem(uniqueId, configurationItemId, description, itemType, configuration, timestamp, userId);
        configurationItems[uniqueId] = newItem;
    }
}
```

7. Conclusions and Future Work

This study proposed a blockchain-based system for security configuration management, which is an enabler of the complete lifecycle management of IT/OT systems, especially in the domain of distributed and large-scale ICT systems. The existing standards, policies, handbooks, guidelines, and best practices were presented, which set the baseline for a robust framework for keeping such systems secure, especially in the context of tracking changes or baseline mismatches in operational environments.

However, although security configuration management is technically well embedded and exercised in current ICT operational environments, according to our research, current IT service management tools are not based on blockchain implementations; moreover, the existing academic research on this topic is very limited. Our approach intends to cover this identified gap, which could potentially underpin the efficiency of configuration management by using the proposed permissioned blockchain model. The proposal of a permissioned model was based on the fact that security and asset management information in such critical systems can only be shared inside a restrictive environment, where controlled access, confidentiality, and need-to-know principles play a critical role. By using such a blockchain model, configuration management can be exercised more effectively and efficiently by inheriting the advantages of distributed ledger technology, such as data integrity, confidentiality, fault tolerance, traceability, transparency, auditability, consistency, and controlled access management, thus supporting the security objectives of organizations.

On the other hand, while the application of blockchain-based models—and especially permissioned ones—grows, it is essential to evaluate the key performance properties of each platform before applying it in real use cases. This gap creates a challenging area of research that requires conducting more technical analyses of blockchain platforms regarding their performance, scalability, and applicability while taking blockchains' inherent limitations into consideration.

The focus of this study is on presenting a theoretical blockchain-based model of a service knowledge management system (SKMS) with a configuration management system (CMS) as a sub-element and on showing how the functions of those elements can benefit from the advantages of a permissioned ledger while interacting with the rest of the system management tools to underpin the objectives of the configuration management process. The end goal is to accurately identify gaps between the "As-Certified" ("As-Should-Be") status and the "As-Maintained" ("As-Is") status of the configuration items of the system (or system of systems) and further facilitate change control, accounting, verification, and auditing, which constitute the main functions of the CM process. This gap analysis is very critical for the security posture of an organization, since via this process, security risks can be easily identified, which could span across many activities, such as malicious activities, unauthorized changes, misconfigurations, incorrect baselining, etc. In addition, the proposed model is enabled by a role-based access control mechanism for smart contracts, as well as by a digital signature function that enables the non-repudiation of system changes.

Our proposal takes place on a theoretical basis, while the future intention is to build the proposed model in a proof-of-concept environment for performance testing, analysis, benchmarking, and evaluation of the operational utility.

Further research can be conducted to integrate deep learning and forecasting algorithms into the blockchain-based system to more accurately identify and prioritize the most important security gaps, as well as to assess how these findings can be integrated into existing risk management and analysis tools for informed decision-making processes.

Author Contributions: Conceptualization, D.C.; Investigation, D.C.; Resources, D.C.; Writing—original draft, D.C.; Writing—review and editing, K.R.; Supervision, K.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. ENISA. *Threat Landscape 2021*; Technical report; ENISA: Athens, Greece, 2021.
2. NIST. *SP 800-37 Revision 2: Risk Management Framework for Information Systems and Organizations*; 2018. Available online: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-37r2.pdf (accessed on 2 March 2023).
3. NIST. SP 800-128: Guide for Security-Focused Configuration Management of Information Systems. 2011. Available online: https://csrc.nist.gov/publications/detail/sp/800-128/final (accessed on 2 March 2023).
4. Joint Task Force Transformation Initiative Interagency Working Group. *Security and Privacy Controls for Federal Information Systems and Organizations*; Technical Report NIST Special Publication (SP) 800-53, Rev. 5, Includes updates as of 12 October 2020; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020. [CrossRef]
5. NIST. SP 800-82 Revision 2: Guide to Industrial Control Systems (ICS) Security. 2015. Available online: https://csrc.nist.gov/publications/detail/sp/800-82/rev-2/final (accessed on 2 March 2023).
6. Rajasekaran, A.S.; Azees, M.; Al-Turjman, F. A comprehensive survey on blockchain technology. *Sustain. Energy Technol. Assess.* **2022**, *52*, 102039. [CrossRef]
7. Berdik, D.; Otoum, S.; Schmidt, N.; Porter, D.; Jararweh, Y. A survey on blockchain for information systems management and security. *Inform. Process. Manag.* **2021**, *58*, 102397. [CrossRef]
8. *ISO/IEC 20000-1:2018*; ISO/IEC 20000—Information Technology—Service Management. International Organization for Standardization: Geneva, Switzerland, 2018.
9. NIST. SP 800-160v1 Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems. 2016. Available online: https://csrc.nist.gov/publications/detail/sp/800-160/vol-1/archive/2018-03-21 (accessed on 2 March 2023).
10. NIST. *Cybersecurity Framework Version 1.1.*; Technical Report Cybersecurity Framework Version 1.1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2018. [CrossRef]
11. Iashchenko, V.V.; Orlova, E.D. A Model for Evaluating the Quality of the Configuration Management Process in the Energy Sector. In Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), St. Petersburg and Moscow, Russia, 26–29 January 2021; pp. 1895–1897. [CrossRef]
12. ISACA. *COBIT 2019 Framework: Introduction and Methodology*, 2nd ed.; ISACA: Rolling Meadows, IL, USA, 2019.
13. *Department of Defense Handbook: Configuration Management Guidance*; Springer International Publishing: New York City, NY, USA, 2020.
14. Kinkelin, H.; Hauner, V.; Niedermayer, H.; Carle, G. Trustworthy configuration management for networked devices using distributed ledgers. In Proceedings of the NOMS 2018—2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018; pp. 1–5.
15. Košťál, K.; Helebrandt, P.; Belluš, M.; Ries, M.; Kotuliak, I. Management and Monitoring of IoT Devices Using Blockchain. *Sensors* **2019**, *19*, 856. [CrossRef] [PubMed]
16. Alvarenga, I.; Rebello, G.; Duarte, O. Securing configuration management and migration of virtual network functions using blockchain. In Proceedings of the NOMS 2018—2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018; pp. 1–9. [CrossRef]

17. Mylrea, M.; Gourisetti, S.N.G. Blockchain for Supply Chain Cybersecurity, Optimization and Compliance. In Proceedings of the 2018 Resilience Week (RWS), Dencer, CO, USA, 20–23 August 2018; pp. 70–76. [CrossRef]
18. Han, S.H. Blockchain-based Configuration Management System. In Proceedings of the 2022 IEEE/ACIS 7th International Conference on Big Data, Cloud Computing, and Data Science (BCD), Danang, Vietnam, 4–6 August 2022; pp. 224–228. [CrossRef]
19. Samaniego, M.; Deters, R. Virtual Resources & Blockchain for Configuration Management in IoT. *J. Ubiquit. Syst. Pervas. Netw.* **2018**, *9*, 1–13.
20. Sarwar, M.I.; Abbas, Q.; Alyas, T.; Alzahrani, A.; Alghamdi, T.; Alsaawy, Y. Digital Transformation of Public Sector Governance With IT Service Management–A Pilot Study. *IEEE Access* **2023**, *11*, 6490–6512. [CrossRef]
21. Riascos Castaneda, R.; Ostrosi, E.; Majić, T.; Stjepandić, J.; Sagot, J.C. A Method to Explore Product Risk in Product Lifecycle Management of Configured Products. *Proc. Des. Soc. Des. Conf.* **2020**, *1*, 687–696. [CrossRef]
22. Wertel, S. *IpX*; Institute for Process Excellence: Denver, CO, USA, 2020.
23. Barrios, P.; Danjou, C.; Eynard, B. Literature review and methodological framework for integration of IoT and PLM in manufacturing industry. *Comput. Ind.* **2022**, *140*, 103688. [CrossRef]
24. Lim, K.Y.H.; Zheng, P.; Chen, C.H. A state-of-the-art survey of Digital Twin: Techniques, engineering product lifecycle management and business innovation perspectives. *J. Intell. Manuf.* **2020**, *31*, 1313–1337. [CrossRef]
25. Bagozi, A.; Bianchini, D.; Rula, A. Multi-perspective Data Modelling in Cyber Physical Production Networks: Data, Services and Actors. *Data Sci. Eng.* **2022**, *7*, 193–212. [CrossRef]
26. Xiong, M.; Wang, H. Digital twin applications in aviation industry: A review. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 1–16. [CrossRef]
27. Bolshakov, N.; Badenko, V.; Yadykin, V.; Celani, A.; Fedotov, A. Digital twins of complex technical systems for management of built environment. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *869*, 6. [CrossRef]
28. Zhang, Q.; Zheng, S.; Yu, C.; Wang, Q.; Ke, Y. Digital thread-based modeling of digital twin framework for the aircraft assembly system. *J. Manuf. Syst.* **2022**, *65*, 406–420. [CrossRef]
29. Le, T.V.; Hsu, C.L. A systematic literature review of blockchain technology: Security properties, applications and challenges. *J. Internet Technol.* **2021**, *22*, 789–802.
30. Sargent, C.S.; Breese, J.L. Blockchain Barriers in Supply Chain: A Literature Review. *J. Comput. Inform. Syst.* **2023**, 1–12. [CrossRef]
31. Karumba, S.; Sethuvenkatraman, S.; Dedeoglu, V.; Jurdak, R.; Kanhere, S.S. Barriers to blockchain-based decentralised energy trading: A systematic review. *Int. J. Sustain. Energy* **2023**, *42*, 41–71. [CrossRef]
32. Pongnumkul, S.; Siripanpornchana, C.; Thajchayapong, S. Performance analysis of private blockchain platforms in varying workloads. In Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, Canada, 31 July–3 August 2017; pp. 1–6.
33. Nasir, Q.; Qasse, I.A.; Abu Talib, M.; Nassif, A.B. Performance analysis of hyperledger fabric platforms. *Secur. Commun. Netw.* **2018**, *2018*, 3976093. [CrossRef]
34. Dabbagh, M.; Choo, K.K.R.; Beheshti, A.; Tahir, M.; Safa, N.S. A survey of empirical performance evaluation of permissioned blockchain platforms: Challenges and opportunities. *Comput. Secur.* **2021**, *100*, 102078. [CrossRef]
35. Monrat, A.A.; Schelén, O.; Andersson, K. Performance evaluation of permissioned blockchain platforms. In Proceedings of the 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Gold Coast, Australia, 16–18 December 2020; pp. 1–8.
36. Melo, C.; Oliveira, F.; Dantas, J.; Araujo, J.; Pereira, P.; Maciel, R.; Maciel, P. Performance and availability evaluation of the blockchain platform hyperledger fabric. *J. Supercomput.* **2022**, *78*, 12505–12527. [CrossRef]
37. Honar Pajooh, H.; Rashid, M.A.; Alam, F.; Demidenko, S. Experimental Performance Analysis of a Scalable Distributed Hyperledger Fabric for a Large-Scale IoT Testbed. *Sensors* **2022**, *22*, 4868. [CrossRef]
38. Wen, Y.F.; Hsu, C.M. A performance evaluation of modular functions and state databases for Hyperledger Fabric blockchain systems. *J. Supercomput.* **2023**, *79*, 2654–2690. [CrossRef]
39. Al-Sumaidaee, G.; Alkhudary, R.; Zilic, Z.; Swidan, A. Performance analysis of a private blockchain network built on Hyperledger Fabric for healthcare. *Inform. Process. Manag.* **2023**, *60*, 103160. [CrossRef]
40. Capocasale, V.; Gotta, D.; Perboli, G. Comparative analysis of permissioned blockchain frameworks for industrial applications. *Blockchain Res. Appl.* **2023**, *4*, 100113. [CrossRef]
41. Cruz, J.P.; Kaji, Y.; Yanai, N. RBAC-SC: Role-based access control using smart contract. *IEEE Access* **2018**, *6*, 12240–12251. [CrossRef]
42. Rouhani, S.; Deters, R. Blockchain based access control systems: State of the art and challenges. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Thessaloniki, Greece, 14–17 October 2019; pp. 423–428.
43. Sookhak, M.; Jabbarpour, M.R.; Safa, N.S.; Yu, F.R. Blockchain and smart contract for access control in healthcare: A survey, issues and challenges, and open issues. *J. Netw. Comput. Appl.* **2021**, *178*, 102950. [CrossRef]
44. Kamboj, P.; Khare, S.; Pal, S. User authentication using Blockchain based smart contract in role-based access control. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 2961–2976. [CrossRef]
45. Zhang, L.; Li, B.; Fang, H.; Zhang, G.; Liu, C. An Internet of Things Access Control Scheme Based on Permissioned Blockchain and Edge Computing. *Appl. Sci.* **2023**, *13*, 4167. [CrossRef]

46. Yang, L.; Jiang, R.; Pu, X.; Wang, C.; Yang, Y.; Wang, M.; Zhang, L.; Tian, F. An access control model based on blockchain master-sidechain collaboration. *Cluster Comput.* **2023**, 1–21. [CrossRef]
47. Fang, W.; Chen, W.; Zhang, W.; Pei, J.; Gao, W.; Wang, G. Digital signature scheme for information non-repudiation in blockchain: A state of the art review. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 1–15. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.