Blockchain-Based Multi-Signature System for Critical Scenarios

Cristina Alcaraz[a], Davide Ferraris[b], Hector Guzman and Javier Lopez[c]
Computer Science Department, University of Malaga,
Campus de Teatinos s/n, 29071, Malaga, Spain

Keywords: Multi-Signature, Blockchain, Trust, and Critical Infrastructures.

Abstract: Blockchain technology plays a crucial role in securing and streamlining transactions across various critical domains. For that reason, this paper presents a Blockchain-based multi-signature system designed for high-stakes scenarios, where both user and Blockchain-generated signatures are required to authorize transactions. By integrating smart contracts, multi-signature coordination, and Blockchain validation, the proposed architecture enhances security, accountability, and resilience. The framework is applied to two key sectors: Mobility and energy. In mobility, it addresses two distinct use cases: Ambulance services, where secure and verifiable authorization of emergency access is required, and insurance claim processing, ensuring transparent, tamper-proof validations. In the energy sector, the system facilitates decentralized, trust-enhanced peer-to-peer energy trading by guaranteeing transaction integrity and compliance. The architecture leverages smart contracts to enforce transaction policies, aggregate multi-signatures, and validate operations while maintaining transparency and reliability. This work highlights the importance of decentralized decision-making and immutable records in securing critical infrastructures. Future research will focus on optimizing performance and evaluating the system's integration with existing Blockchain platforms such as Ethereum and Hyperledger.

1 INTRODUCTION

In the evolving landscape of Blockchain technology, security and control over digital assets remain paramount concerns (Narayanan, 2016). Multi-signature (or multisig) mechanisms, which require multiple parties to sign a transaction before it can be executed, have emerged as a robust solution to enhance security and trust (Buhler, 2025). Traditionally, multisig implementations involve multiple user signatures to authorize a transaction (Roy and Karforma, 2012). However, an innovative paradigm introduces the Blockchain itself as an active participant in the signing process, thereby broadening the scope and utility of *multi-signature schemes* (Buhler, 2025) and its applicability to heterogeneous scenarios in which the criterion of transparency is a primary requirement. Thus, this paper explores the concept of multisig in a Blockchain context where Blockchain technology operates as one of the signatories. This approach integrates the decentralized nature of Blockchain systems with user-centric controls to secure transactions (Zhang et al., 2025). The Blockchain's signature can be viewed as a form of automated governance or programmatic approval, where predefined rules and conditions are met before the system's implicit authorization is granted (Saurabh et al., 2024). By incorporating the Blockchain as a signatory, a novel interplay is created between system automation and user authorization (Kuznetsov et al., 2024). For example, the Blockchain may enforce rules such as compliance with Smart Contract (SmC) conditions, verification of user identity through decentralized identity protocols, or adherence to community-governed policies. Once these criteria are satisfied, the user must confirm the transaction with their signature to complete the process. This model not only strengthens transaction security but also ensures a higher degree of transparency and accountability. Thus, the integration of Blockchain-signed multi-signature schemes introduces several potential applications, including secure multi-party computation, Decentralized Finance (DeFi) transactions, cross-chain interoperability, and enhanced fraud prevention mechanisms (Dohler et al., 2024). Additionally, it opens possibilities for regulatory compliance where automated checks can ensure adherence to legal requirements before the user authorizes the final step (Zhuk, 2025). For all these reasons,

[a] https://orcid.org/0000-0003-0545-3191
[b] https://orcid.org/0000-0003-3035-3774
[c] https://orcid.org/0000-0001-8066-9991

Alcaraz, C., Ferraris, D., Guzman, H. and Lopez, J.
Blockchain-Based Multi-Signature System for Critical Scenarios.
DOI: 10.5220/0013559000003979
In *Proceedings of the 22nd International Conference on Security and Cryptography (SECRYPT 2025)*, pages 221-232
ISBN: 978-989-758-760-3; ISSN: 2184-7711

the *main contribution of this paper* is (i) to provide an analysis of the theoretical foundations, (ii) the technical implementation and (iii) the practical implications of Blockchain-assisted multi-signature systems, especially for those deployed in certain critical contexts (Alcaraz et al., 2011; Lopez et al., 2013) such as energy and mobility. In order to achieve this task, we will delve into the core principles of multisig, the role of the Blockchain as a transaction signatory, and potential challenges and opportunities associated with this paradigm. The paper also reveals a relevant set of requirements, which are key to the design of the approach with applicability in critical scenarios, as well as a validation methodology based on the "matching" of such requirements to operations (first by layers of functionality and then by services). The idea is to provide the literature with the mechanism by which (i) to demonstrate the usefulness of the approach in critical scenarios and (ii) to identify which services of the approach should be optimized, considered or even prioritized in the future.

The paper is organized as follows. Section 2 adds the related work to provide later on the requirements we want to satisfy with our solution in Section 3. In Section 4, we provide the general design of the proposed architecture and how it is matching the proposed requirements is presented; whereas the mapping among the proposed architecture, requirements and functionalities to use case scenarios related to the mobility and energy sectors is analyzed in Section 5. In the following Section 6, we present in more detail each of the scenarios, introducing the role of the JavaScript Object Notation (JSON) and a Solidity code implemented for the presented use cases. In section 7, the paper concludes and outlines future work.

2 RELATED WORK

Multi-signature schemes have been widely studied and applied in various Blockchain contexts, contributing to enhanced security and functionality. This section summarizes key advancements in this field based on significant contributions from existing literature. For example, Lin *et al.* (Lin et al., 2019) introduced a Blockchain-based mobile, ticketing system that leverages SCs and multisig to securely execute and authorize transactions. Their system ensures the authenticity and security of mobile tickets through Blockchain verification and the use of an immutable ledger. The proposed approach demonstrates high efficiency with minimal costs, positioning it as a viable solution for secure and cost-effective ticketing systems. Also, Boneh *et al.* (Boneh et al., 2018) developed novel multisig schemes designed to reduce the size of the Bitcoin Blockchain while maintaining versatility for other multisig applications. Their schemes support signature compression and public-key aggregation, enabling verifiers to authenticate a multi-party signature using a compact representation. Additionally, they introduced an Accountable Subgroup Multisignature (ASM) scheme, where the signature size is independent of the number of signers, enhancing scalability and practicality for applications such as Bitcoin multisig addresses.

Xiao *et al.* (Xiao et al., 2020) focused on improving the efficiency of transactions in enterprise Blockchain platforms by proposing two multisig schemes: Gamma MultiSignature (GMS) and Advanced Gamma MultiSignature (AGMS). Their schemes are designed to address the complexities and inefficiencies of traditional multisig processes. Through implementation on Hyperledger Fabric, they demonstrated that AGMS achieves high transaction efficiency, low storage complexity, and robust security against rogue-key and k-sum problem attacks, making it a strong candidate for enterprise-level applications. Also, Aitzhan *et al.* (Aitzhan and Svetinovic, 2016) addressed the challenge of securing transactions in decentralized smart grid energy trading systems. They proposed a Blockchain-based solution incorporating multisigs and anonymous encrypted messaging to enable secure and private energy trading without relying on trusted third parties. Their proof-of-concept demonstrated robust security and privacy mechanisms while ensuring efficient performance in decentralized energy markets.

Kara *et al.* (Kara et al., 2023) introduced a multisig scheme based on RSA aimed at reducing Blockchain size and improving resistance to known attacks. Their scheme operates in the plain public key model, simplifying implementation by avoiding the need for key possession proofs. The proposed ASM model discloses the subset of signers responsible for a valid signature and employs a two-round protocol for public-key aggregation. This scheme enhances Blockchain efficiency and security while maintaining scalability. Similarly, Gai *et al.* (Gai et al., 2022) proposed a Blockchain-based Multi-Signature Lock (BMSL-UAC) to address security challenges in the metaverse's Ubiquitous Access Control (UAC) settings. Their scheme enables secure and traceable access to data in consortium Blockchain systems, ensuring that only authorized users can interact with the system. The experimental evaluation on Hyperledger demonstrated that their approach achieves reasonable performance in terms of resource consumption, delay, and throughput, making it a suitable framework

for managing access control in the metaverse.

The body of work discussed here highlights the versatility and innovation in multi-signature applications, ranging from ticketing systems and energy trading to Blockchain compression and access control. These advancements collectively inform the development of Blockchain-integrated multi-signature schemes and their potential to address challenges in security, efficiency, and scalability. Based on these works and their progress, and particularly in the field of Blockchain and multi-signature, we now explore in the following section how to go one step beyond the state of the art by providing a Blockchain-based multi-signature architecture. The architecture has the proposal of combining Blockchain and multisig under the pragmatic vision of integrating four layers of functionality. This perspective not only simplifies the use of the technique in decentralised systems, but also favours its application in heterogeneous and dynamic contexts where it is relevant to intensify principles of accountability, but also to provide guarantees of modularity and performance.

3 REQUIREMENTS

Any Blockchain-based multi-signature system with application in critical scenarios must satisfy a minimum set of security, efficiency and reliability requirements in order to generate trust and better use of its utility. In that regard, this section establishes the key requirements, categorized into: (i) *Blockchain Infrastructure-specific Requirements* (IR) and its deployment in real contexts; (ii) the *Multisignature mechanism Requirements* (MR) for trust and accountability management; and (iii) *Application Requirements* (AR) associated with the final use of Blockchain for real-world scenarios. Considering these three sets of requirements and based on (Alcaraz and Lopez, 2012; Alcaraz et al., 2020), we now identify a subset of control and trust conditions. Starting with the deployment of the Blockchain infrastructure and attending the features of a Distributed Ledger Technology (DLT), IR comprises:

- *Traceability* (IR-1): Transactions must be fully traceable, allowing all stakeholders to verify the flow of assets or approvals. Blockchain provides an immutable record of signatures and transaction steps by design, ensuring visibility throughout the system.
- *Immutability* (IR-2): Once recorded, transactions cannot be altered or deleted. This ensures that past signatures and approvals remain intact, preventing unauthorized modifications to digital records.
- *Verifiability* (IR-3): All transactions should be independently verifiable by any stakeholder. Cryptographic proofs and publicly accessible Blockchain records allow third parties to confirm the validity of transactions without reliance on a central authority.
- *Sustainability* (IR-4): Energy-efficient cryptographic techniques and Blockchain protocols should be considered to minimize the environmental impact of transaction processing. Proof-of-Stake (PoS) or delegated PoS (DPoS) Blockchains may provide more sustainable alternatives (Saad et al., 2021).

Trust can also be managed through the multisig system integrated into Blockchain. The resulting system adheres the beneficial technical characteristics to intensify the properties of trust, such as the use of hashes and signatures. Thus, within MR, we consider:

- *Accountability* (MR-1): Each participant of the Blockchain network must be accountable for their actions, particularly in multisig transactions where multiple parties sign a transaction. The Blockchain ensures that each transaction is explicitly linked to authorized signers.
- *Auditing* (MR-2): DLTs should provide a verifiable record of all transactions for auditing and compliance purposes. The distributed ledger of the Blockchain offers a transparent and automated accounting mechanism, reducing the risk of fraud.
- *Lightweight* (MR-3): Given resource constraints, particularly in mobile and (Industrial) IoT environments (Alcaraz and Lopez, 2014), the multisig mechanism should be computationally efficient. Optimized cryptographic algorithms and minimal on-chain storage requirements ensure a lightweight design.
- *Non-repudiation* (MR-4): A signer cannot later deny their participation in a transaction. Cryptographic signatures stored on the Blockchain provide undeniable proof of participation, ensuring that all actions are verifiable.

Regarding AR and its related requirements for the protection of critical infrastructures and the preservation of their performance when using Blockchain-based multisig approaches:

- *Performance* (AR-1): DLTs must support fast transaction processing to enable real-time applications such as mobility and energy trading. SC optimizations and off-chain scaling solutions (Gupta et al., 2023) may be necessary to meet this requirement.

- *Decoupling* (AR-2): DLTs should minimize dependencies between different components, enabling flexible integration with various Blockchain networks and external services. This allows seamless interoperability with multiple applications and industries.
- *Resilience* (AR-3): DLTs must withstand malicious attacks, software bugs, and external disruptions. By decentralizing control and using fault-tolerant consensus algorithms, Blockchain ensures resilience against system-wide failures, in addition to guaranteeing the availability and accessibility of the data at all times.
- *Survivability* (AR-4): DLTs must remain operational even in the presence of attacks, failures, or disruptions. Thus, Blockchain should ensure the management of multiple copies of transactions, enhancing the system's resilience, under suitable immutability and authentication approaches.

All these requirements will not only form the basis of the general design of the architecture proposed below, but will also lay the foundations for the construction of future multisig approaches. In this case, the special combination of Blockchain and a multisig strategy adds a "*trust wrapper*" of at least two digital signatures: (i) one performed by the Blockchain, which acts as a third trusted entity, and (ii) another by the participant(s), who is the owner of the transaction.

4 LAYERED ARCHITECTURE

To comply with IR, MR and AR, multisig approaches must not only consider the DLT as a key element within their construction, but also associate security services according to layers of functionality:

1. *Application Layer* (AL): Handles user interactions and interfaces, where all the requirements application-level should widely be considered.
2. *Smart Contract Layer* (SCL): Manages business logic, rules, and transaction authorization criteria.
3. *Multi-Signature Coordination Layer* (MSCL): Guarantees the correct signature both by the user and by the Blockchain.
4. *Blockchain Layer* (BL): Verifies and records transactions and participates as an elementary signatory within the proposed architecture.

In turn, these layers are divided into subcomponents of functionality in order to (i) modularize services and (ii) intensify operational performance, such that AL deals with:

- *User Interface (UI)*: A web or mobile application enabling users to initiate transactions, view status, and provide their signatures.
- *Client Wallets* (CW): Secure wallets where users store their private keys to sign transactions.
- *Transaction Module* (TM): Allows users to create transactions by specifying details (e.g., recipient, amount).

On the contrary, SCL contains the following three sub-components:

- *Transaction Validation Contract* (TVC): Defines rules for when the Blockchain can authorize a transaction (e.g., compliance checks, preconditions). It also verifies that all required conditions are met before the Blockchain signs.
- *Multi-Signature Execution Contract* (MSEC): Coordinates the process of collecting both the user and Blockchain's signatures. It also ensures atomic execution, where transactions are only valid if both parties sign.

About MSCL, it is based on three subcomponents:

- *Key Management Module* (KM): Generates and stores the Blockchain's private key securely (e.g., HW security modules or threshold cryptography).
- *Signature Aggregator* (SA): Collects user and Blockchain signatures and combines them into a single multisig.
- *Transaction Verifier* (TV): Validates the aggregated signature before broadcasting the transaction to the Blockchain.

Finally, the BL contains:

- *Consensus Mechanism* (CM): Ensures decentralized agreement on the validity of transactions before the Blockchain signs. Participates as a signatory by generating a Blockchain-level signature using a system-controlled private key.
- *Immutable Ledger* (IL): Records transactions, their signatures, and metadata for auditing and transparency.

To clarify the workflow taken by the multisig procedure within the approach, the user first initiates a transaction in AL, corresponding to *(«stage-1»)* in Figure 1. The details of the transaction are sent to SCL *(«stage-2»)*, where a Blockchain validation process checks predefined rules (e.g., thresholds, compliance, conditions). All entities involved in the signature process interact with the SCL, resulting in the generation of a SC transaction that serves as verifiable evidence of the signature. At the MSCL *(«stage-3»)*, all collected signatures are validated, and

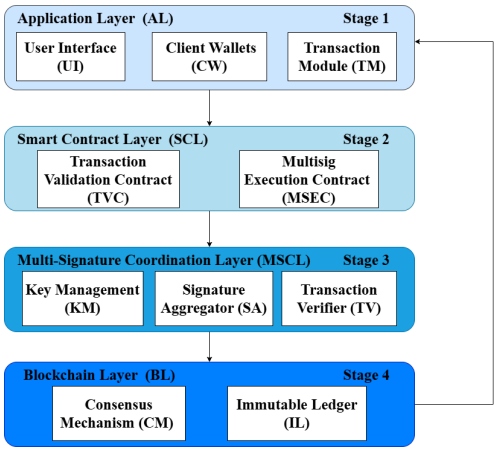

Figure 1: Layered Blockchain-based multisig.

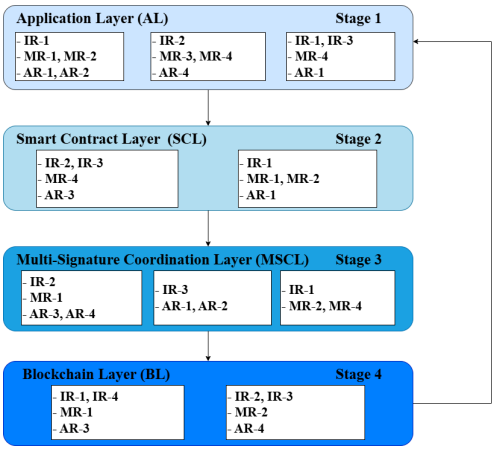

Figure 2: Mapping requirements to functionality layers.

a blockchain-based signature is appended. This signature does not represent a traditional cryptographic signature but the deterministic outcome of the SC validation process. It ensures the validity of the entire procedure, is publicly verifiable, and remains immutable. Once finalised this process, the system requires user verification by sending the transaction back to the user for final confirmation and signature using his/her private key. After this stage, the procedure for adding multiple signatures begins. The signature aggregator combines the Blockchain and user signatures into a single valid multisig, and the aggregated signature is appended to the transaction. The following stage is dedicated to the transmission of the transaction to the Blockchain, *«stage-4»*. This means that the finalized transaction, with its multisig, is sent to BL for consensus and inclusion in the ledger. This feature allows the possibility for audit and feedback, because the immutable ledger records the transaction for future reference. As the final stage in the workflow of the proposed approach, the system returns to *«stage-1»* to notify the user. To achieve this task, AL informs the user about the status of the transaction. As shown in the figure, the architecture emphasizes modularity and scalability, but also the security of the user and the organizations involved in the application. Each layer is responsible for a specific activity and interacts with the user to provide guarantees of non-repudiation and trust. The mapping of these security properties with the requirements of Section 3 is analyzed in the following section to validate the approach as a whole and to show its final utility.

4.1 Mapping Requirements to Architecture Layers

The Blockchain-based multisig system proposed in Section 4 is now extended to map the requirements established in Section 3 to functionality layers. The idea is to verify that all the requirements are widely addressed in accordance with the critical characteristics of the application context. Precisely, Figure 2 characterizes the aforementioned assignment and highlights how the different layers of the architecture can work together to fulfill critical aspects. In fact, by verifying that IR, MR and AR are covered by specific subcomponents, we can also be more confident that the system will be able to meet the necessary conditions of trust, transparency and usability.

More in details, we consider for the A-L subcomponents the following requirements:

- UI: Must allow users to track their transactions and verify approvals (IR-1); each user action (transaction initiation, approval) should be linked to an identifiable entity (MR-1); UI should provide access to past transaction logs for compliance verification (MR-2); UI must offer real-time transaction status updates with minimal latency (AR-1); and UI should be flexible enough to support different Blockchain networks (AR-2).
- CW: Ensures that private keys remain unchanged and cannot be altered maliciously (IR-2); wallets should be optimized for low-resource environments such as mobile or (I)IoT scenarios (MR-3); CW guarantees that once a signature is applied, the user cannot deny signing (MR-4); and CW must support key recovery mechanisms for resilience against user key loss (AR-4).
- TM: Transactions must be logged with identifiable metadata (IR-1); users should be able to confirm the details before submission (IR-3); TM should prevent incorrect or incomplete transactions from being signed (MR-4); and TM should allow seamless transaction creation without unnecessary delays (AR-1).

About the SC-L requirements, each subcomponent guarantees the following:

- TVC: Once deployed, contract rules cannot be altered to ensure consistency (IR-2); SC execution should be provable and independently auditable (IR-3); transactions should only proceed if all necessary conditions are cryptographically validated (MR-4); and the contract must be resistant to tampering and external attacks (AR-3).
- MSEC: Approvals must be logged transparently on-chain (IR-1); each transaction should include a record of both user and Blockchain signatures (MR-1); ensures that transactions cannot be executed unless all required signatures are present, *Ipso Facto* (MR-2); and signature aggregation must be optimized for minimal gas costs (AR-1).

Thirdly, for the MSC-L, the matching requirements for each subcomponent are:

- KM: Secure storage prevents unauthorized key modifications (IR-2); cryptographic key ownership must be provable and auditable (MR-1); private keys must be protected from attacks and hardware failures (AR-3); and KM must implement backup and recovery mechanisms (AR-4).
- SA: Ensures that all aggregated signatures are correct and tamper-proof (IR-3); SA must efficiently combine multiple signatures without excessive computational overhead (AR-1); SA should support different multisig schemes and cryptographic standards (AR-2)
- TV: Must log verification steps for future audits (IR-1); TV has to verify the multi-signatures against stored public keys to confirm authenticity (MR-2); and TV ensures that an approved transaction cannot be disputed (MR-4).

As for the requirements of the B-L subcomponents are as follows:

- CM: Must ensure that the transaction lifecycle is fully recorded and verifiable (IR-1); it should optimize resource usage while maintaining security (IR-4). Moreover, each validated transaction should be linked to a responsible signing entity (MR-1); and CM should be resistant to Sybil attacks and collusion (AR-3);
- IL: Ensures that once recorded, transactions cannot be altered or removed (IR-2); IL must allow all stakeholders to independently confirm transaction authenticity (IR-3); IL should support forensic analysis and compliance auditing (MR-2); and IL ensures transaction data remains available even in case of network failures (AR-4).

If off-chain processing techniques are additionally implemented outside the chain, it is also possible to improve performance and scalability, reducing transaction costs and delays. All this also indicates that the matching shown in Figure 2 can serve as an attractive tool to guide future multi-signature approaches.

5 MAPPING TO SCENARIOS

After exploring the capabilities of DLTs and the multi-signature service for trust, it is important to consider how the approach is applied in real-world scenarios. Thus, the following subsections focus on delving into some examples of how these technologies can be utilized in different scenarios (hereon as SC).

5.1 SC1 - Mobility

In mobility applications, DLT can be used for congestion avoidance, traffic safety, car leasing or selling, parking services, and insurance. Data shared by vehicles can even facilitate the development of intelligent traffic lights for smart city approach (Elassy et al., 2024). Energy management is another crucial area where Blockchain can play a significant role in mobility scenarios, particularly with the rise of electric and autonomous vehicles (Rana et al., 2024). Proposed applications include managing access to charging stations and controlling battery cycles in autonomous cars. Likewise, Blockchain's immutability also allows for the secure recording of events related to vehicle use, such as driver identity and visited locations, which can aid in the development of smart public transport systems (Jabbar et al., 2022). Blockchain's payment capabilities can also be used in the mobility sector for services, such as payment at charging stations and SC-based rental car platforms (Dey et al., 2024). However, traditional database systems currently surpass Blockchains in terms of performance due to their distributed nature and the immense quantity of data produced by the context of application itself, for example for consumption, user data, charging, traffic, control. The best-known current Blockchain applications are capable of processing a very low number of transactions compared to non-distributed systems. Reducing this latency is therefore necessary to achieve compliance with AR-1 (Jabbar et al., 2022). In order to verify the applicability of the approach proposed throughout this paper, we extend the scenario to two particular use cases for mobility: (i) The first one is related to *ambulance coordination*, and the (ii) the second one to i*nsurance claim processing*.

5.1.1 SC1a: Ambulance Coordination

In this first case, a Blockchain-based multi-signature system can be used for ambulance dispatch and route authorization. The system ensures secure, transparent, and real-time decision-making among emergency responders, hospitals, and traffic control authorities. In this subsection, we consider how the requirements presented above are prioritized according to the characteristics of this scenario and its level of criticality. This means that the availability of control and coordination operations takes precedence over other security measures such as authentication or confidentiality. For the sake of simplicity and space, we limit the study to those requirements that have a higher or medium impact in the context of application.

- *Performance (AR-1) → High Priority*: Ambulance dispatch requires real-time approval of emergency routes, ensuring low-latency transaction validation. The Blockchain must support fast execution of SCs to approve emergency passages through restricted areas.
- *Decoupling (AR-2) → High Priority*: The Blockchain should be able to integrate with multiple health providers and regulatory frameworks in order to be available for all the different actors involved in such scenario.
- *Survivability (AR-4) → High Priority*: The system must operate even during network failures to guarantee continuous emergency service availability. Decentralized architecture ensures multiple nodes retain transaction data, preventing data loss.
- *Verifiability (IR-3) → High Priority*: Authorities (police, hospitals) must be able to verify route approvals without relying on a central entity. The system should provide tamper-proof proof of authorization to prevent fraud.
- *Accountability (MR-1) → Medium Priority*: Every signed transaction (e.g., dispatch approval, route modification) should be linked to an identifiable entity.
- *Resilience (AR-3) → Medium Priority*: The system should be fault-tolerant against cyber-attacks or system overloads during crises.

5.1.2 SC1b: Insurance Claim Processing

In this second case, we can state that a Blockchain multisig system can be implemented for automating and verifying insurance claim processing. SCs validate accident reports, driver liability, and policy coverage, ensuring secure and transparent processing. As discussed for the first case, we now explore the requirements and priorities for the scenario.

- *Non-Repudiation (MR-4) → High Priority*: Drivers, insurance companies, and law enforcement must sign transactions related to accident reports. A cryptographic proof of signatures ensures no party can later deny involvement.
- *Traceability (IR-1) → High Priority*: Every step of the claim process (accident reporting, damage assessment, claim approval) should be logged and verifiable. Prevents fraud by ensuring immutable transaction history.
- *Auditing (MR-2) → High Priority*: Regulators and auditors must be able to verify all claims to prevent fraudulent insurance payouts.
- *Decoupling (AR-2) → Medium Priority*: The system should be flexible to integrate with different insurance providers and governmental authorities.
- *Performance (AR-1) → Medium Priority*: While real-time performance is not as critical as in emergency response, the system must process claims efficiently to reduce delays.

5.2 SC2: ENERGY

As the focus on the renewable energy sources has increased, the energy market has also shifted into a distributed market where renewable energy is traded. Due to this effect, the number of Blockchain-based solutions designed for the energy sector has grown in the last years (Ma et al., 2024). In addition, carbon emission trading systems and green certificates rely on Blockchain attributes, such as transparency and immutable data recording, to establish a reliable market (Prawitasari et al., 2024). In this scenario, Blockchain can be useful to control the decentralized grid and effectively solve the problem of control the output power reasonably to avoid unstable voltage on the grid produced by the excess of power in different nodes of the grid. Smart contracts can also be a solution to detect the real power consumption of users and evaluate the agreed contracts, automatically giving incentives or punishments to the users (Su et al., 2024). Nevertheless, despite the leverage of recording the data consumption in the distributed ledger for the study of the real consumption, the public nature of the Blockchain brings privacy concerns about consumer's daily activity patterns (Joshi et al., 2018). Another approach is the implementation of distributed auction systems permit buyers and sellers complete reliable, safe and transparent auctions. Moreover, the auction payments process can be automatized by the deployment of smart contracts and

electricity flows can be detected to verify that the transactions are completed successfully (Chitra et al., 2023). Back to the privacy concerns, a distributed energy system based on tokens is proposed (Mehdinejad et al., 2022). Therefore, consumers can negotiate electricity prices anonymously, protecting their personal information during the transactions. To sum up, Blockchain's transparency and immutable record benefit carbon emission trading and green certificates and also offers solutions for grid control, power consumption monitoring, and decentralized energy auctions. However, privacy concerns arise due to the public nature of Blockchain (Hewa et al., 2021). As addressed for mobility, we now discuss how the requirements presented earlier can be prioritized in the energy sector case. In this case, a Blockchain-based multi-signature system is deployed for energy trading and decentralized grid management. The system facilitates peer-to-peer energy transactions, while guaranteeing secure, auditable and sustainable operations without a significant impact on control. Here, the considered requirements are the following.

- *Sustainability (IR-4) → High Priority*: Blockchain technology should use energy-efficient consensus mechanisms to align with green energy goals.
- *Resilience (AR-3) → High Priority*: The system must withstand cyberattacks or technical failures to maintain continuous energy distribution. Moreover, the decentralized grid must operate even if individual nodes fail.
- *Immutability (IR-2) → High Priority*: Energy transactions must be tamper-proof to prevent fraud or unauthorized modifications.
- *Auditing (MR-2) → High Priority*: Regulators and consumers should be able to verify transactions for transparency in energy pricing and distribution.
- *Performance (AR-1) → Medium Priority*: While real-time execution is not as critical as in mobility, energy trading systems should still process transactions efficiently to avoid delays.
- *Decoupling (AR-2) → Medium Priority*: The Blockchain should be able to integrate with multiple energy providers and regulatory frameworks.

5.3 Final Discussions

Previous studies state that the most relevant requirements for critical scenarios (SC1a and SC2) are those related to AR-1, AR-2 and AR-3 (see Table 1), and therefore related to performance, decoupling and resilience, as also stated in (Alcaraz and Lopez, 2012); whereas for the rest of scenarios (and beyond AR-1, AR-2 and AR-3), MR-2 (audit) is the most prominent. In addition, looking at the most prominent requirements, we also note that the majority fall under the "AR" requirement class, which once again indicates that the type and nature of the scenario are fundamental when designing DLT-assisted approaches. In fact, the deployment of DLT and its respective solutions must be solutions that help improve the operational functions of each scenario, but not become a burden that affects the performance and effectiveness of those scenarios. Therefore, for SC1a and SC2, and prioritizing their critical nature, it is recommended that the following services be optimized for future designs and implementations.

Table 1: Identifying relevant requirements at layer level.

App	AR-1	AR-2	AR-3	AR-4	MR-1	MR-2	MR-4	IR-1	IR-2	IR-3	IR-4
SC1a	X	X	X	X	X					X	
SC1b	X	X				X	X	X			
SC2	X	X	X			X			X		X

Now, considering AR-[1-3] and MR-2, Table 2 shows the subcomponents or services of the Blockchain and the multisig systems that should be optimized in relation to them, giving priority to those related to the user interface, followed by MSEC and SA. This result is quite reasonable, since the main use of the approach is to show its multisig capacity based on DLT. The approach could lose its real usefulness if it does not provide attractive interfaces to manage the signing capabilities, and accessbility to the end user.

Table 2: Identifying relevant requirements at service level.

Reqs	UI	TM	TVC	MSEC	KM	SA	TV	IM	IL
AR-1	X	X		X		X			
AR-2	X					X			
AR-3			X		X			X	
MR-2	X			X			X		X

Beyond this theoretical demonstration, in the following section we show the usefulness of the approach from a practical point of view and for the three use case scenarios (SC1a, SC1b, SC2).

6 PRACTICAL VIEW

This section provides the JSON and Solidity codes to integrate and demonstrate the real applicability of the approach for automating multi-signatures for critical

scenarios. The solidity codes are available at Github[1], because for space limitations we only provide here the description and a portion of the code. On the other hand, we have divided this section into three chief subsections: (i) one devoted to SC1a on emergency services and its coordination, (ii) SC1b on insurance management, and (iii) SC2 for energy management

6.1 SC1a: Ambulance Coordination

As mentioned above, emergency medical services can apply Blockchain systems with multiple signatures to authorize ambulance movement through restricted zones (e.g., toll roads, traffic lights, restricted lanes). The hospital, traffic authority, and Blockchain system must approve the route before execution.

6.1.1 JSON for Contextual Conditions

Listing 1 represents a piece of JSON code about an ambulance requesting access to restricted roads, with multiple signatures required before authorization. Basically, the JSON represents a structured request for an emergency vehicle to access restricted routes. In this system, an ambulance submits a request containing a unique identifier, along with details about its origin and destination, specifying the planned route with multiple checkpoints. Each checkpoint requires approval from three key entities: the hospital initiating the request, the traffic authority overseeing road access, and the Blockchain system that ensures compliance and security. The transaction remains in an "Awaiting Signatures" state until all required parties have signed off, ensuring that only authorized emergency vehicles receive clearance. Thus, by leveraging the Blockchain-based multisig mechanism, it guarantees that all approvals are traceable, immutable, and verifiable, facilitating real-time decision-making for emergency responses.

Listing 1: JSON for Ambulance Scenario.

```
{"request_id": "AMB_12345",
 "ambulance_id": "AMB_001",
 "hospital_id": "HOSP_789",
 "route":[{"checkpoint": "Toll Road 23",
           "status": "Pending"},
          {"checkpoint": "Restricted Lane A5",
           "status": "Pending"}],
 "signatures":{"hospital": false,
               "traffic_authority": false,
               "blockchain": false},
 "status": "Awaiting Signatures",
 "timestamp": "2025-02-26T12:00:00Z"}
```

[1]https://github.com/ferrarisUMA/SECRYPTPaper

6.1.2 Solidity for Smart Contract

In the Github, an example of a Solidity SmC is found, implementing a multi-signature mechanism for the ambulance coordination scenario; and Listing 2 illustrates a portion of such a specification.

Listing 2: Solidity Code for Ambulance Scenario (Portion).

```
Contract AmbulanceAuthorization{
    address public hospital;
    address public trafficAuthority;
    address public blockchainAuthority;
    address public ambulance;
    struct RouteRequest{
        string requestId; string ambulanceId;
        bool hospitalApproved; bool trafficApproved;
        bool blockchainApproved; bool executed;}
    mapping(string => RouteRequest) public requests;
    event RouteRequested(string requestId,
                         string ambulanceId);
    event RouteApproved(string requestId,
                        address approver);
    event RouteExecuted(string requestId);}
    //For +info: In our Github
```

The developed Solidity SmC for this scenario is designed to facilitate and regulate emergency vehicle access in urban environments. This contract defines a structure for emergency access requests, each containing an ambulance ID, route details, and the required multi-signatures from the key entities showed before in the JSON code: the hospital, the traffic authority, and the Blockchain itself. The contract ensures that a transaction remains pending until all necessary parties sign it, guaranteeing compliance and security. Once the required signatures are collected, the transaction is executed, granting the ambulance access to restricted routes. The SmC also maintains a log of all approved requests, ensuring traceability and accountability. The use of Blockchain and multisig validation eliminates unauthorized access while enabling swift, automated clearance for emergencies.

6.2 SC1b: Insurance Claim Processing

Here, we consider how our approach can be useful after a car accident, where drivers, police officers, and insurance companies must verify the claim before compensation is processed. The Blockchain ensures non-repudiation, traceability, and transparency.

6.2.1 JSON for Contextual Conditions

Listing 3 represents an insurance claim request in JSON where multiple signatures (from driver, police, and insurer) are needed to approve the claim. The JSON code defines an automated process for han-

dling vehicle accident claims. When a driver submits a claim, the request is linked to a unique ID and includes supporting details such as accident location, timestamp, and involved parties. Verification is a multi-step process that requires digital signatures from the driver, a police officer validating the incident, and the insurance company that must approve compensation. The Blockchain ensures that no claim can be altered or repudiated, preventing fraud and ensuring accountability. The transaction is marked as "Awaiting Signatures" until all necessary approvals are recorded. Once fully signed, the claim is executed and stored on the Blockchain, providing a transparent and tamper-proof history of insurance transactions.

Listing 3: JSON for Insurance Scenario.

```
{"claim_id": "CLAIM_98765",
 "details": "Details of the event",
 "driver_id": "DRV_456",
 "police_report":{"officer_id": "POL-789",
                  "status": "Pending"},
 "insurance_approval": {"insurer_id": "INS-123",
                        "status": "Pending"},
 "blockchain_verification":{"verified": false},
 "status": "Awaiting Signatures",
 "timestamp": "2025-02-26T12:00:00Z"}
```

6.2.2 Solidity for Smart Contract

Both Listing 4 and the aforementioned Github contain the corresponding SmC (in Solidity), developing a multisig mechanism for the insurance claims processing scenario. The Solidity SmC automates and secures the vehicle accident claim process. When a claim is initiated, the contract registers key details such as the claimant's identity, accident specifics, and the claim amount. The claim must then be approved by three parties: the driver (who submits the claim), the police (who verifies the accident's occurrence), and the insurance company (which authorizes the compensation). The multisig mechanism ensures that all required entities validate the claim before any payout is processed, preventing fraudulent submissions and enforcing accountability. Once all signatures are obtained, the contract finalizes the claim, releasing funds to the claimant and immutably storing the transaction on the Blockchain. This decentralized approach enhances transparency, reduces processing time, and eliminates disputes over claim legitimacy.

Listing 4: Solidity Code for Insurance Scenario (Portion).

```
Contract InsuranceClaim{
    address public driver;
    address public police;
    address public insurer;
    struct Claim{
        string claimId;
        bool policeApproved; bool insurerApproved;
        bool blockchainVerified; bool executed;}
    mapping(string => Claim) public claims;
    event ClaimSubmitted(string claimId,
                         string driver);
    event ClaimApproved(string claimId,
                        address approver);
    event ClaimProcessed(string claimId);}
    //For +info: In our Github
```

6.3 SC2: Energy

In this use case, we consider a peer-to-peer energy trading system where households with solar panels sell excess energy to neighbors using a Blockchain-based platform ensuring secure and transparent transactions. Thus, a household (seller) initiates an energy trade by specifying the amount of energy and price. A neighbor (buyer) agrees to the terms. The Blockchain validates the trade against predefined rules (i.e. energy availability, grid capacity). It also checks that the buyer's payment is deposited into a SmC escrow. Once validated, the Blockchain signs the transaction. Then, after the energy is transferred, the buyer confirms receipt by signing the transaction. Following this scheme, we assure that the signatures of both the Blockchain and buyer are aggregated, authorizing the payment to the seller. For auditing, the immutable ledger logs the transaction for regulatory and billing purposes. Moreover, this approach increases transparency and accountability in energy markets.

6.3.1 JSON for Contextual Conditions

Listing 5 illustrates the corresponding JSON code, representing a decentralized energy transaction where a user buys energy from a peer-to-peer energy trading platform, applying the multisig concept for transaction authorization. Specifically, the JSON structure characterizes an energy producer, such as a solar panel owner, selling surplus electricity to a consumer. The producer initiates the transaction by specifying the amount of energy available and the price per unit. The consumer then places a purchase request, and the Blockchain validates the agreement before executing the trade. Before completion, the transaction requires multisig approvals from the producer, consumer, and the Blockchain system to ensure compliance with sustainability and accountability standards. The entire transaction process is recorded immutably, providing a transparent and auditable history of energy exchanges. Through Blockchain and multisig verification, this approach guarantees trust, reduces reliance on centralized intermediaries, and enhances efficiency in decentralized energy markets.

Listing 5: JSON for Ennergy Scenario.

```
{"transaction":{
    "id":"energyTxn7890",
    "buyer":{
        "id":"user123", "name":"John",
        "WalletAddress":"0xBuyerWalletAddress"},
     "seller":{
        "id":"producer456", "name":"SolarFarm Inc.",
        "WalletAddress":"0xSellerWalletAddress"},
     "energyDetails":{
        "quantity":"50 KWh",
        "princePerUnit":"0.02 ETH",
"multiSigature":{
       "requiredSignatures": 2,
       "signatures":[
        {"signer":"0xPlatformWalletAddress",
        "Signature":"0xSignatureFromPlatform"},
        {"signer":"0xBuyerWalletAddress",
        "Signature":"0xSignatureFromBuyer"}]},
"status": "Completed",
"timestamp": "2025-02-08T14:30:00Z"}}
```

6.3.2 Solidity for Smart Contract

Both Listing 6 and the Solidity SmC include the code parts, which integrates the multisig approach for trading energy between a buyer and a seller through a Blockchain. The SmC enables a decentralized peer-to-peer energy market where producers and consumers engage in trustless transactions. The contract records offers from producers specifying energy availability and pricing, allowing consumers to place purchase requests. A transaction is executed only when three signatures are provided: the producer, the consumer, and the Blockchain, which verifies compliance with predefined regulations (such as sustainability standards or maximum trading limits). The contract guarantees that energy trades are fair, transparent, and immutable. Upon completion, the energy transfer is recorded permanently on the Blockchain, allowing for auditing and regulatory oversight. Certainly, the multisig mechanism ensures that neither party can manipulate the transaction, creating a secure and efficient energy marketplace without the need for centralized intermediaries.

Listing 6: Solidity Code for Energy Scenario (Portion).

```
Contract EnergyTradingMultiSig{
    struct EnergyTransaction{
        address buyer; address seller;
        uint256 quantity; uint256 pricePerUnit;
        bool platformSigned; bool buyerSigned;
        bool Completed;}
    mapping(uint256 => EnergyTransaction)
            public transaction;
    address public platform;
    uint256 public transactionCounter;
    event TransactionCreated(uint256 transactionId,
                        address buyer,
                        address seller,
                        uint256 quantity,
                        uint256 pricePerUnit);
    event TransactionSigned(uint256 transactionId,
                        address signer);
    event TransactionCompleted(uint256 transactionId,
                        address seller);
    constructor(address_platform){platform=_platform;}
    modifier onlyPlatform(){
        required(msg.sender==platform,
        "Only plataform can perform this action");}
    function createTransaction(address_buyer,
        address_seller, uint256_quantity,
        uint256_pricePerUnit)}
```

7 CONCLUSIONS

This paper presented a Blockchain-based multisig system designed to enhance security, transparency, and trust in critical scenarios. By integrating user confirmations with Blockchain-generated signatures, the proposed architecture has four layers and ensures several key requirements such as performance, resilience, and auditing in transaction processing. Such system was applied to two key sectors: Mobility and Energy trading. The ambulance coordination use case demonstrated how the framework enables real-time decision-making in emergency services, while the insurance claim processing use case highlighted its role in fraud prevention and transparent claim verification. Additionally, the energy trading scenario showcased how the architecture supports decentralized energy markets. Future work will focus on evaluating the architecture across different Blockchain platforms, such as Ethereum, Solana, and Hyperledger, to identify the most suitable solution in terms of efficiency, scalability, and regulatory compliance.

ACKNOWLEDGEMENTS

This work has been partially supported by two EU projects under GA No: 101086308 (DUCA, HORIZON-MSCA-2021-SE-01) and 101131292 (AIAS, HORIZON-MSCA-2022-SE-01); and by the two following national projects: 5G+TACTILE_4 (NEXTGENERATION.UE, Spanish UNICO 5G I+D) under Grant TSI-063000-2021-26, and SECAI funded by the MCIN/AEI 10.13039/501100011033 and FSE+ under the Grant PID2022-139268OB-I00.

REFERENCES

Aitzhan, N. Z. and Svetinovic, D. (2016). Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE transactions on dependable and secure computing*, 15(5):840–852.

Alcaraz, C., Agudo, I., nez, D. N., and Lopez, J. (2011). Managing incidents in smart grids à la cloud. In *IEEE CloudCom 2011*, pages 527–531, Athens, Greece. IEEE Computer Society.

Alcaraz, C. and Lopez, J. (2012). Analysis of requirements

for critical control systems. *International journal of critical infrastructure protection*, 5(3-4):137–145.

Alcaraz, C. and Lopez, J. (2014). WASAM: A dynamic wide-area situational awareness model for critical domains in smart grids. *Future Generation Computer Systems*, 30:146–154.

Alcaraz, C., Rubio, J. E., and Lopez, J. (2020). Blockchain-assisted access for federated smart grid domains: Coupling and features. *Journal of Parallel and Distributed Computing*, 144:124–135.

Boneh, D., Drijvers, M., and Neven, G. (2018). Compact multi-signatures for smaller blockchains. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 435–464. Springer.

Buhler, M. (2025). Enhancing multi-signature cryptocurrency wallets with risk-based authentication.

Chitra, T., Ferreira, M. V., and Kulkarni, K. (2023). Credible, optimal auctions via blockchains. *Cryptology ePrint Archive*.

Dey, A. K., Gope, B., and Mandal, B. K. (2024). Applications of blockchain for future mobility. In *Blockchain Technology in the Automotive Industry*, pages 147–157. CRC Press.

Dohler, M., Lopez, D. R., and Wang, C. (2024). *Blockchains in 6G: A Standardized Approach to Permissioned Distributed Ledgers*. CRC Press.

Elassy, M., Al-Hattab, M., Takruri, M., and Badawi, S. (2024). Intelligent transportation systems for sustainable smart cities. *Transportation Engineering*, page 100252.

Gai, K., Wang, S., Zhao, H., She, Y., Zhang, Z., and Zhu, L. (2022). Blockchain-based multisignature lock for uac in metaverse. *IEEE Transactions on Computational Social Systems*, 10(5):2201–2213.

Gupta, A., Gupta, R., Jadav, D., Tanwar, S., Kumar, N., and Shabaz, M. (2023). Proxy smart contracts for zero trust architecture implementation in decentralised oracle networks based applications. *Computer Communications*, 206:10–21.

Hewa, T. M., Hu, Y., Liyanage, M., Kanhare, S. S., and Ylianttila, M. (2021). Survey on blockchain-based smart contracts: Technical aspects and future research. *IEEE Access*, 9:87643–87662.

Jabbar, R., Dhib, E., Said, A. B., Krichen, M., Fetais, N., Zaidan, E., and Barkaoui, K. (2022). Blockchain technology for intelligent transportation systems: A systematic literature review. *IEEE Access*, 10:20995–21031.

Joshi, A. P., Han, M., and Wang, Y. (2018). A survey on security and privacy issues of blockchain technology. *Mathematical foundations of computing*, 1(2).

Kara, M., Laouid, A., and Hammoudeh, M. (2023). An efficient multi-signature scheme for blockchain. *Cryptology ePrint Archive*.

Kuznetsov, O., Sernani, P., Romeo, L., Frontoni, E., and Mancini, A. (2024). On the integration of artificial intelligence and blockchain technology: a perspective about security. *IEEE Access*, 12:3881–3897.

Lin, K.-P., Chang, Y.-W., Wei, Z.-H., Shen, C.-Y., and Chang, M.-Y. (2019). A smart contract-based mobile ticketing system with multi-signature and blockchain. In *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, pages 231–232. IEEE.

Lopez, J., Alcaraz, C., and Roman, R. (2013). Smart control of operational threats in control substations. *Computers & Security*, 38:14–27.

Ma, C.-Q., Lei, Y.-T., Ren, Y.-S., Chen, X.-Q., Wang, Y.-R., and Narayan, S. (2024). Systematic analysis of the blockchain in the energy sector: Trends, issues, and future directions. *Telecommunications Policy*, 48(2):102677.

Mehdinejad, M., Shayanfar, H., and Mohammadi-Ivatloo, B. (2022). Decentralized blockchain-based peer-to-peer energy-backed token trading for active prosumers. *Energy*, 244:122713.

Narayanan, A. (2016). *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press.

Prawitasari, P. P., Nurmalasari, M. R., and Kumalasari, P. D. (2024). Blockchain technology in the carbon market: Enhancing transparency and trust in emissions trading. *Jurnal Revenue: Jurnal Ilmiah Akuntansi*, 5(2):1495–1521.

Rana, M. T., Numan, M., Yousif, M., Hussain, T., Khan, A. Z., and Zhao, X. (2024). Enhancing sustainability in electric mobility: Exploring blockchain applications for secure ev charging and energy management. *Computers and Electrical Engineering*, 119:109503.

Roy, A. and Karforma, S. (2012). A survey on digital signatures and its applications. *Journal of Computer and Information Technology*, 3(1):45–69.

Saad, S. M. S., Radzi, R. Z. R. M., and Othman, S. H. (2021). Comparative analysis of the blockchain consensus algorithm between proof of stake and delegated proof of stake. In *2021 International Conference on Data Science and Its Applications (ICoDSA)*, pages 175–180. IEEE.

Saurabh, K., Rani, N., and Upadhyay, P. (2024). Towards novel blockchain decentralised autonomous organisation (dao) led corporate governance framework. *Technological Forecasting and Social Change*, 204:123417.

Su, X., Hu, Y., Liu, W., Jiang, Z., Qiu, C., Xiong, J., and Sun, J. (2024). A blockchain-based smart contract model for secured energy trading management in smart microgrids. *Security and Privacy*, 7(1):e341.

Xiao, Y., Zhang, P., and Liu, Y. (2020). Secure and efficient multi-signature schemes for fabric: An enterprise blockchain platform. *IEEE Transactions on Information Forensics and Security*, 16:1782–1794.

Zhang, C., Liao, W., Liu, X., Wu, H., and Alenazi, M. J. (2025). A multi-signature scheme for defending malleability attack on defi. *IEEE Access*.

Zhuk, A. (2025). Beyond the blockchain hype: addressing legal and regulatory challenges. *SN Social Sciences*, 5(2):1–37.