Adaptive Fuzzy Logic Risk- Based Access Control Model for Smart Contract Execution on Block Chain Systems

Omondi, Alan Odhiambo[1], Erick Oteyo Obare[2], Samuel Oonge[3]
[1,2,3]*Department of Information Technology, School of Computing and Informatics, Maseno University, Kisumu, Kenya*

***Abstract*: Smart contracts contribute to the automation and efficiency of various processes, reducing the need for intermediaries in order to execute agreements on the blockchain platforms. Classical traditional access control models, Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role-Based Access Control (RBAC) represent conventional access control paradigms which have played a fundamental role in the management of resource access in many organizations and systems. These approaches employ predefined policies and conventions to govern and enforce access permissions. The access models of smart contracts deployed in Blockchain systems exhibit limited adaptability to dynamic changes in the system environment. Conventional crypto, the main access control mechanism encounters a challenge in effectively mitigating the security risks related to identity management authentication across the processes of consensus, initiation, and execution of smart contracts, specifically within Blockchain systems. The shortcoming of classical access control models, which were established in previous times lie in their lack of adaptability and responsiveness in detecting abnormal and malevolent behaviors through the process of observing and tracking user actions during the entirety of their access session. The situation at hand necessitates the implementation of adaptive access control models. The aim of this paper is to propose the creation of a fuzzy logic risk-based access control model that is both dynamic and adaptive. The study will approach the proposed model creation, testing and evaluation by adopting a Mixed-Method research design which includes Experimental research design. Action research design will be used to test and evaluate the model anchored on within the PiECE framework. A fuzzy logic inference principle with expert judgment technique will be employed to evaluate the model through an evaluation metric criterion using regression model analysis. To handle uncertain data ranges, encompassing categories such as severe, high, moderate, and low user risk estimating strategy based on fuzzy logic shall be employed. Data collection methods will utilize the Ai data mining technique route thereafter involve cleaning, pre-processing and annotation of the sample size data sets. The cleaned data will be split into training, testing and validating data sets which then empirically, the MATLab toolkit will be used in the development and testing phase of the proposed architecture for execution stages of smart contracts in blockchain platform. Ethical concerns shall be highlighted based on the pilot model's efficacy. The attributes of the adaptive fuzzy logic access control model will be utilized in future to design intelligent contracts that dynamically adjust the capabilities of users' based on their behaviors throughout access sessions to enhance further smart contracts' inherent secured nature.**

***Keywords*: Security risk. Fuzzy logic. Fuzzification. Logical Inference. Defuzzyfication. Fuzzy operators. Fuzzy set. Membership function (MF). Expert judgment mechanism.**

1. Introduction

A. Background of the Study

A blockchain refers to a decentralized and distributed digital ledger, documenting transactions across numerous computers in a secure and transparent manner (Bankykoom et al.,2018). It comprises a series of blocks, each containing transaction records. Central attributes of blockchain systems encompass decentralization, immutability, transparency, and security. Unlike traditional centralized systems, blockchain is decentralized to operate on a peer-to-peer network of computers (nodes). Each node on the network has a copy of the entire blockchain, and there is no central authority controlling the system (Arslan et al.,2020). This decentralization helps enhance security and resilience. Transactions are grouped together in blocks, and each block contains a unique identifier called a hash, a timestamp, and a reference to the previous block's hash. This creates a chain of blocks, hence the term "blockchain." To agree on the state of the blockchain and validate transactions, blockchain networks use consensus mechanisms. Common ones include Proof of Work (used by Bitcoin), Proof of Stake, Delegated Proof of Stake, and others. These mechanisms guarantee that the nodes in the network obtains a consensus on the validity of transactions. Once a block is added to the blockchain, it is extremely difficult to alter or delete the information within it (Tilson et al.,2017).

Manuscript revised November 28, 2024; accepted November 29, 2024. Date of publication November 30, 2024.
This paper available online at www.ijprse.com
ISSN (Online): 2582-7898; SJIF: 5.59

This is due to the cryptographic hash functions used to link blocks and the consensus mechanisms that make it computationally infeasible to alter historical transactions. History of the entire transaction is visible to all participants in the network. Anyone with access to a blockchain node can view the complete record of transactions. However, the level of privacy can vary depending on the specific blockchain and its design. Many blockchain systems are associated with cryptocurrencies. For example, Bitcoin operates on a blockchain and has its native cryptocurrency (BTC). Other blockchains, like Ethereum, support the creation of various tokens, including their native cryptocurrency (Ether, or ETH), (Castiglione et al.,2016).

Some applications of blockchains, like Ethereum, support smart contracts. Smart contracts are agreements with terms directly coded into them, which automatically execute and enforce these terms when specific conditions are fulfilled (Buterin et al., 2018). Operating on a blockchain, smart contracts facilitate trustless and decentralized automation of processes, eliminating the necessity for intermediaries. Smart contracts are written in programming languages specifically designed for the blockchain platform they run on. For example, Code Execution, Ethereum uses Solidity. The code of a smart contract is deployed to the blockchain. Smart contracts operate on a decentralized blockchain network. The code and execution are distributed across multiple nodes, making the process resistant to censorship or interference from a single party. They automatically execute when predefined conditions specified in the code are met (Bankykoom et al., 2018). This removes the need for a third party to enforce or validate the terms of the contract and as such need for a trustee. The trust is established through the code and the decentralized consensus mechanism of the blockchain (Watanabe et al.,2016). Once deployed, then the code of a smart contract becomes immutable. This ensures that the terms and conditions agreed upon in the contract remain unchanged and can be relied upon. The code and execution of smart contracts are transparent and visible on the blockchain. Participants can verify the contract's status, terms, and outcomes at any time (Aitzhan et al.,2016). Inherently smart contracts have a wide range of applications, including financial services (e.g., decentralized finance or DeFi), supply chain management, voting systems, insurance and so on. Smart contracts may rely on external information to trigger actions. Oracles, which are external data sources, can be integrated to provide real-world data to smart contracts (Azbeg et al.,2021).

Access control models forms an integral part of smart contracts' computing resources, which serve to manage and monitor access within a system. The access control model fall into three primary categories: classical access control models, dynamic access models, and object-based access models. Classical models, such as MAC, and RBAC, rely on predefined rules, while dynamic models like DAC, AAC, and UBAC consider dynamic factors for access decisions. Object-Based Access Control (OBAC) focuses on individual objects, allowing fine-grained control but necessitating complex implementation (Dolgui et al.,2020). Despite their strengths, classical models like MAC, RBAC, and ABAC have limitations, such as lack of centralized control or fine-grained access. Dynamic models are more intricate to manage, involving numerous attributes and policies. Current access models struggle with security concerns in blockchain systems, especially during smart contract execution, as they lack flexibility and sensitivity to abnormal actions (Aitzhan et al., 2016).

While smart contracts offer numerous advantages, they also face several challenges that need to be addressed for broader adoption and improved functionality. Security is a critical challenge associated with executing smart contracts on a blockchain platform. Vulnerabilities and bugs in the smart contract's code could lead to exploits hence the need to conduct code audits, formal verification, and rigorous testing which are essential to mitigate these risks. The immutability of smart contracts, while a strength in terms of trust, becomes a challenge if there are bugs or vulnerabilities in the deployed code (Aitzhan et al., 2016). Once deployed, fixing such issues is difficult, and it requires careful consideration during the development phase (Ruddick et al.,2018). Many blockchain networks, especially those with high transaction volumes like Ethereum, face scalability challenges. As the number of transactions and smart contracts increases, the network may experience congestion and slower transaction processing times. Smart contracts on one blockchain may not be directly compatible or interoperable with those on another blockchain. This lack of standardization can hinder collaboration and limit the potential for integrated applications across different platforms (Huang et al.,2020).

The legal and regulatory status of smart contracts is still evolving. Legal and regulatory challenges may arise as smart contracts operate in a somewhat novel legal and regulatory landscape. Ambiguities in legal frameworks and uncertainties about the enforceability of smart contracts in traditional legal systems can pose challenges, especially in cross-border transactions (Hwang et al.,2020). Smart contracts often rely on external data sources (oracles) to trigger actions based on real-world events. The reliability and security of oracles are crucial, as inaccurate or manipulated data can lead to incorrect contract executions. Developing and understanding smart contracts often requires a deep understanding of blockchain technology and programming languages specific to the platform. Improving the user-friendliness of smart contract development tools is essential for broader adoption (Arslan et al.,2020). While blockchain transactions are transparent, privacy concerns may arise when executing smart contracts that involve sensitive or private information. Solutions such as zero-knowledge proofs are being explored to address these concerns. Updating or upgrading a deployed smart contract is challenging due to its immutability. Developers must consider mechanisms for introducing changes while ensuring backward compatibility and minimizing disruptions.

B. Problem Statement

Smart contract utilization has brought unlimited benefits, but at the same time raises several security issues. This is because current access control models with rigid, inflexible and static structure with predefined rules that always give the same result in different situations fail to provide the required level or degree of security for such execution rendering system. The main gap lies in the lack of dynamism and adaptability within existing smart contract blockchain access models, which predominantly use cryptography as their classical access control mechanism. These traditional approaches are insufficient for detecting malicious actions or protecting system resources once access is granted. Classical access control approaches do not provide a way to detect malicious actions and protect system resources after granting the access. Consequently, if an abnormal action is detected, user privileges cannot be appropriately reduced, nor can the access session be effectively terminated. The risk estimation module used in dynamic access control model has no flexibility to adjusts a user's permission adaptively depending on user's behaviour in active access sessions such that if an abnormal action is discovered, user privileges will be reduced to some degree or the access session will be terminated.

2. Literature Review

A. Introduction

This chapter provides an overview of the literature reviewed for the theoretical foundations of the dissertation. Initially, we delve into literature related to the theoretical, conceptual, and frameworks concerning blockchain and smart contract security paradigms. We scrutinize literature pertaining to the security risks associated with smart contracts on blockchain technology.

A significant focus of the literature review is on the incorporation of emerging disruptive technologies and the theoretical paradigms guiding empirical research in creating an adaptive risk-based access control model for the execution of smart contracts. This model aims to address the security checksum for users before they are granted access and control within a fully integrated smart contract blockchain architecture.

B. Theoretical Framework

The theoretical underpinning of this dissertation draws from two primary research streams: information systems software platforms and access software development approaches.

1) Information System Software Platforms: Apache Cordova

In line with the principles of the Apache Software Foundation (ASF), the Cordova application framework is widely utilized by developers for creating applications. This platform offers tools and interfaces that facilitate the development of applications published across multiple platforms. Cordova supports various platforms, including Android, iOS, Windows, Ubuntu, Blackberry 10, WP8, and OS X. Key features include the Web View for user-friendly interfaces, Web App for configuration settings, and Cordova Plugin for seamless communication within application components and the platform.

The Apache Cordova architecture and governance mechanisms emphasize proactive shaping and evolution of products to ensure future relevance in terms of dynamism, flexibility, and adaptability. Existing software development approaches are explored within the framework of control and prediction, which highlights their focus on positioning software in an exogenous environment for future relevance (Firdhous et al.,2018).

2) Smart Contracts Management Tool and Fuzzy Logic Toolkit

Data extraction for our study involves utilizing the smart contracts management tool, focusing specifically on a dataset comprising issues raised by active contributors. This dataset includes information such as requests for information, bug fixes, feature requests, and suggestions. We choose smart contracts describing specific feature requests or issues with the application and platform. Smart contracts specifically in freight and logistics and legal environments are particularly suitable for this research as they provide transactional details, including issue descriptions and implemented solutions using blockchain technology within the application.

In summary, the literature review forms the theoretical foundation for the dissertation, incorporating insights from information systems software platforms and access software development approaches (Huang et al., 2018). The Apache Cordova framework is explored as a key element, emphasizing proactive development strategies, and data extraction from smart contracts management tools is outlined for further analysis using a fuzzy logic toolkit that is integral part of the classical access control module as shown in figure 2.1a and figure 2.1b below.

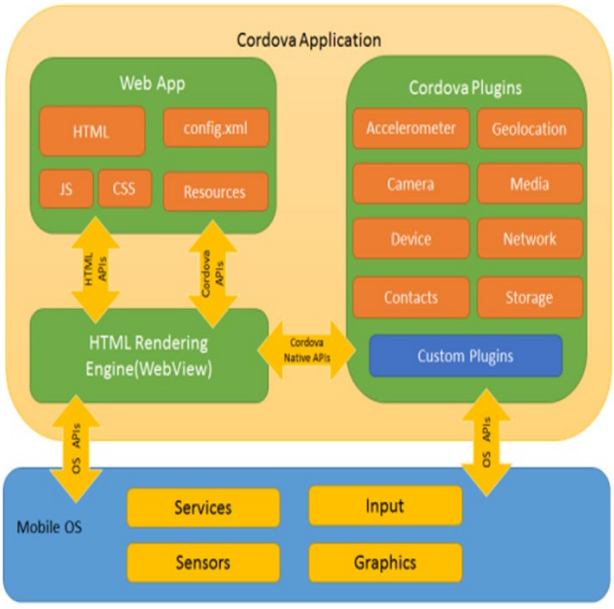

Fig. 1. 1a: Information system software platforms the apache cordova architecture (huang et al., 2018).

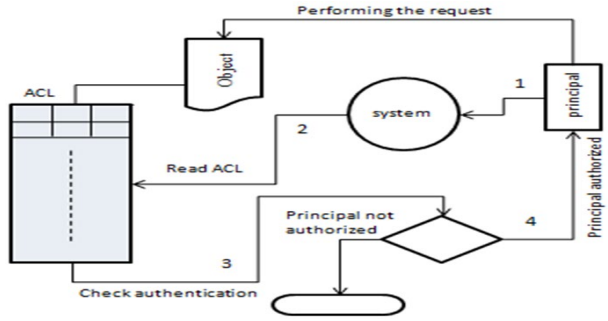

Fig. 2. 1b: Classsical access control module within the apache cordova architecture (kortesniemi et al.,2014).

3) Framework of Control and Prediction

These theoretical framework approaches are derived from the architecture and governance mechanisms of the platform ecosystem, necessitating a proactive shaping and evolution of the product to maintain future relevance (Zhang et al., 2015). To tackle these challenges, we investigate the fundamental principles of current software development approaches through the lens of the control and prediction framework, particularly within the dynamic access control module as depicted in Figure 2.2 (Zhang et al., 2017). The framework illustrates that prevailing software development strategies are centered on situating the software product in an exogenous environment to ensure its relevance, usability, security, and dynamic utility.

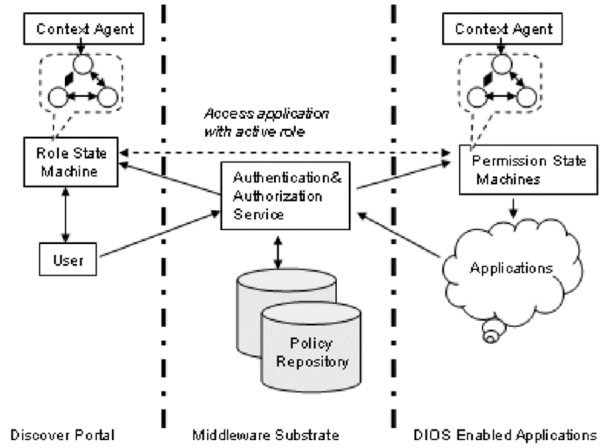

Fig. 3. Dynamic access control module (Zhang et al., 2017).

C. Exploratory Review on Risk-Based Estimation Paradigms

1) Game Theory

Game theory, regarded as a branch of applied mathematics, finds application in various domains such as evolutionary biology, economics, artificial intelligence, political science, and information security. Its purpose is to delineate decision scenarios involving multiple participants, conceptualized as games where each player makes strategic choices to maximize their payoff while anticipating reasonable actions from opponents. In interactive environments, game theory serves as a primary tool for modelling and constructing automated decision-making processes, offering consistent and mathematical frameworks. The efficacy of game theory lies in its methodology for analyzing strategic choice problems (Binmore et al.,2015).

The game theory modelling process involves the decision-maker interacting with players, understanding their strategic decisions, and observing their preferences and responses. A game theory comprises four essential components: players, strategies, payoffs, and information. Players are the decision-makers within the game, and strategies represent the plans they employ in response to the moves of other players. Selecting suitable tactics is critical for players. Payoffs denote the rewards players receive in the game, influenced by both their actions and those of other players (Binmore et al.,2015). However, a critique of the game theory paradigm notes that risk analysis is based on user benefits rather than probability. Additionally, game theory is recommended in situations where practical data is lacking, and it becomes complex, especially with more than two players. The use of mixed strategies can lead to random outcomes, making it less adaptable in smart contract identification management.

2) Decision Tree

The decision tree is a widely used methodology in various machine learning operations, functioning as a decision support tool that generates decisions based on a set of rules organized in a tree structure. Constructing a decision tree model involves the partitioning of data into training and validation sets. Training data are employed to derive the necessary rules for the tree, while validation data are utilized to assess the tree and implement required modifications. Represented as a flow diagram, the decision tree features nodes, represented by rectangles, each describing the probability and impact of a risk. These rectangles are interconnected by arrows, with each arrow leading to another box indicating the percentage probability.

Decision tree approaches are known for their ease of comprehension and their significance in data classification. They can function effectively with limited data when experts provide all the necessary rules (Santos et al., 2019). Moreover, decision trees present all potential alternatives and paths in a single view, facilitating easier comparison among various options.

Despite its advantages, the decision tree model comes with certain limitations. One such limitation is its scalability, where an increase in the scale of the tree can make the resulting model challenging to interpret, requiring additional data for rule validation. Additionally, the decision tree model is based on expectations, making it difficult to plan for all contingencies that may arise from a decision (Santos et al., 2019).

3) Risk Assessment

Risk assessment is employed to examine potential damages associated with a specific scenario. It can be defined as the systematic investigation of potential security breaches to a system and the resulting losses, utilizing a combination of available information about the situation and informed judgment regarding unknown information. The purpose of risk assessment is to recognize the context of risk and establish

acceptable risk values for each situation, achieved through comparisons with similar risks in analogous scenarios. Additionally, it aims to propose alternative solutions to mitigate risk and evaluate the effectiveness of those solutions. The selection of the appropriate type of risk analysis depends on the available data characterizing the probability and impact of the risk. While an effective risk assessment offers numerous benefits, such as providing a well-founded basis for preventing or minimizing the impact of risk, it is a subjective process influenced by experience and is only valid at a specific point in time, limiting its adaptability in validating user identity in smart contract executions (Cazzola et al.,2018).

A comparison between different risk estimation approaches in terms of usability, time complexity, scalability, flexibility, subjectivity, and computing power requirements is shown in table 1.

It is clear that there is no straightforward dynamic approach that can be used without limitations for risk estimation approach without subjectivity to manage user identification in smart contract execution process. Scalability and adaptability seem to be a problem or gap in most approaches. Therefore, choosing the optimal risk estimation approach should depend heavily on the context and data sets used in a smart contract.

4) Fuzzy Logic Inference Data System Technique

A fuzzy logic inference data system is a computational approach that simulates human thinking by describing the world in imprecise terms. Unlike computers that operate only on precise evaluations, the human brain can engage in reasoning with uncertainties and judgments. The fuzzy logic system is a precise problem-solving approach capable of working with both numerical data and linguistic knowledge simultaneously. It simplifies the management of complex systems without the need for a mathematical description (Atlam et al.,2021).

The computation process using the fuzzy logic system comprises three main phases:

Fuzzification – This phase converts crisp or classical variables of input and output into fuzzy variables to process and produce the desired output. Most variables are initially crisp, and fuzzification is used to handle imprecise information.

Fuzzy Inference Process – This phase involves building IF-THEN fuzzy rules to describe relationships between different inputs and output. Linguistic variables are used to represent conditions and outputs, creating rules that guide the fuzzy logic system in processing input data.

Defuzzification – This phase converts the fuzzy output back to a crisp output since the final result needs to be a precise variable.

Fuzzy sets, incorporated in a black-box approach, excel at handling complex mathematical equations and formulas, making them applicable in various computing modelling applications. Fuzzy logic, conceptualized by Zadeh (1995) provides a convenient way to map an input space to an output space, offering advantages such as modelling imprecise multivariate data and nonlinear functions of arbitrary complexity, based on natural language.

The general concept behind fuzzy logic involves applying a set of pre-defined rules (if-else statements) in parallel to interpret values in the input vector and assign values to the output vector. Fuzzy logic is based on fuzzy sets, where membership is not a simple true-false answer but a not-quite-true-or-false response within the unit interval [0,1]. Fuzzy logic has flexibility, robustness, and ease of understanding due to its basis in natural language. However, it requires domain experts to create accurate rules and involves more tests and simulations, which can be time-consuming, especially with an increasing number of rules (Bai et al.,2016).

D. Evaluation of Risk Estimation Methods and Approaches of Access Controls in Smart Contract Execution

The table 1 and 2 gives a summary evaluation of the risk estimation methods and approaches of access controls in executing smart contracts.

Table 1
Pros and cons of risk estimation paradigms (Atlam et al., 2021)

Risk estimation technique	Benefits					Limitations		
	Usable	Fast	Scalable	Dynamic	Include expert experience	Enormous resources needed	Time overhead	Subjective
Fuzzy logic system	✓			✓	✓		✓	✓
Expert judgment	✓	✓			✓			✓
Risk assessment		✓	✓		✓		✓	✓
Game theory	✓			✓		✓		✓
Decision tree		✓		✓	✓	✓	✓	✓

E. Blockchain Technology Models

Blockchain is a chain structure formed by the orderly concatenation of data blocks according to the generation time, a distributed database with the characteristics of decentralization, collective maintenance, tamper proof, and distrust, which is especially suitable for building a programmable money system (Abdelmaboud et al.,2022). Blockchain technology has been widely used in medical care, finance, Internet of Things, energy, and many other fields. Blockchain can generally be divided into three categories: public blockchain, consortium blockchain and private blockchain according to the access permission. Public blockchains are open to all users in the world, so any user can read data and broadcast transactions on the chain. The consortium blockchains are jointly managed by several business-related institutions, each of which runs one or more nodes, and the read-write permissions are limited to the nodes in the consortium. The read-write permissions of the private blockchains are controlled by an organization, and the qualifications of participating nodes are strictly limited (Ahubele et al.,2021).

A blockchain functions as a decentralized and distributed digital ledger, documenting transactions across numerous computers in a secure and transparent manner (Buterin, 2018). Utilizing digital encryption and distributed consensus algorithms, blockchain establishes a decentralized system of

trust without relying on trusted individual nodes. It operates through a chain of blocks, with each block containing a roster of transactions. The key features of blockchain systems include decentralization, immutability, transparency, and security. Unlike traditional centralized systems, blockchain is decentralized to operate on a peer-to-peer network of computers (nodes). Each node on the network has a copy of the entire blockchain, and there is no central authority controlling the system (Arslan et al.,2020). This decentralization helps enhance security and resilience. Transactions are grouped together in blocks, and each block contains a unique identifier called a hash, a timestamp, and a reference to the previous block's hash. This creates a chain of blocks, hence the term "blockchain." To agree on the state of the blockchain and validate transactions, blockchain networks use consensus mechanisms (Casino et al.,2019). Once a block is added to the blockchain, it is extremely difficult to alter or delete the information within it. This is due to the cryptographic hash functions used to link blocks and the consensus mechanisms that make it computationally infeasible to alter historical transactions. The entire transaction history is visible to all participants in the network (Castiglione et al., 2016).

1) Blockchain Security Attributes to Smart Contracts During Access Session

Blockchain includes technologies such as distributed architectures, consensus algorithms, and smart contracts. Smart contract technology can ensure that users who do not trust each other complete transactions without any third-party trusted intermediaries or authorities. Simultaneously, smart contracts in digital form can be flexibly embedded in various tangible or intangible assets, transactions, and data to realize active or passive assets, information management and control, and gradually build programmable smart assets, systems, and society (Azbeg et al.,2021). In terms of smart contract working mechanism, it contains two attributes: the state variable and state value. In the smart contract program, If-Then and What-If statements are used to set the triggering scenarios and response rules of the terms in the contract. Through multi-party mutual agreement and digital signature, the user submits the transaction initiated. After propagation through the blockchain network and verification by each node, it is stored in blocks of the blockchain. The user obtains the contract address and contract interface, and invokes the contract during trading (Aitken et al., 2016). Miners accept the incentive mechanism set by the system, contribute their computing power to verify transactions, and generate contracts or execute contract codes in the local sandbox after receiving the contract creation or invocation command. The contract codes automatically determine whether the current scenario meets the contract trigger conditions to strictly implement the response rules and update the world state. After the transaction is verified. to be valid, it is packaged into a new data block, which is linked to the main chain of the blockchain after consensus authentication (Bangare et al.,2016). Owing to the differences in blockchain platforms and their operating mechanisms, and the differences in smart contract development languages, the operating mechanisms of smart contracts are also different. Therefore, the working mechanism of the smart contract is explained from the three common aspects of the smart contract subject, the data loading method, and the execution environment (Buterin et al.,2018).

Table 2
Strengths and weakness of risk estimation paradigms (Odhiambo et al., 2024)

Risk Methods and Approaches	Strengths	weakness	Citation
Risk-Adaptable Access Control (RAdAC)	proposes three principles for constructing a risk-based access control model: estimating risk, defining an acceptable risk value, and controlling data distribution based on the acceptable risk value	lacks quantitative evaluation of risk values and real-time features.	Windley et al. (2018)
Benefit and Risk-based Access Control (BRAC)	utilizing security risk and system benefits to decide access measuring security risks quantitatively	lacks flexibility for smart contract security management	Zhang et al. (2017) Diep et al. (2019)
Dynamic risk-based decision approach,	suggest three approaches for risk estimation based on subject trustworthiness and object sensitivity, employing user actions for access decisions	lacking details on quantitative evaluation. it lacks adaptability.	Khan et al. (2017)
Multi-Level Security (MLS) model	measuring risk based on the difference between object and subject security levels	challenges related to scalability, dynamism, and adaptability	Chen et al. (2016)
Game theory for risk analysis	mathematical functions and risk-based models	face challenges in quantitative evaluation and lack real-time contextual features lack real-time contextual information and dynamism, crucial for smart contract environments	Rajbhandari et al. (2016) Binmore et al. (2015)
Decision tree	known for ease of comprehension in data classification. They can function effectively with limited data when experts provide all the necessary rules	limitation is its scalability, where an increase in the scale of the tree can make the resulting model challenging to interpret, requiring additional data for rule validation. Based on expectations, making it difficult to plan for all contingencies that may arise from a decision.	Santos et al. (2019).

At the core of any smart contract lies the fundamental concept of a chain of transactions interconnected through cryptographic signatures, rendering them immutable across networks and decentralized in terms of ownership and control (Chandrasekhar, 2022). In this ledger, transactions are cryptographically linked, ensuring their resistance to tampering, and are shared among connected users. Transactions that are verified within the ledger cannot be altered without consensus from the users involved. While Bitcoin is the most commonly associated application of Blockchain, its utilization has expanded and diversified rapidly in recent years, with a projected growth in its market (Decker et al., 2017).

Blockchain consists of a continually growing ledger of cryptographically signed and immutable transactional records, distributed across all network participants. Each transaction is timestamped and linked to preceding entries, allowing anyone with appropriate access to trace any event back through the entire transaction history, regardless of the participant involved. According to Chandrasekhar (2022), the capabilities of Blockchain encompass the digital representation of assets, facilitation of new forms of value exchanges, decentralized interaction and transactions without the need for a central authority or intermediary. Additionally, Finally, it enables the management, governance, and execution of partnerships and smart contracts across a variety of data entities. Object-oriented access control module has been advocated to address the security checksum, particularly during the initiation of smart contracts within blockchain systems (Gupta et al., 2018). Figure 4 and 5 illustrates the structure, functionality and life cycle of smart contract within blockchain technology.

Smart contracts execution environment consists of two main types of execution environments namely: virtual machines and containers (dockers). The virtual machine and container are similar to a sandbox that isolates and limits the resources used by the contract while executing the contract code (Wu et al.,2018). A virtual machine usually refers to the software implementation of a computer with complete hardware functions that can execute programs like a real machine, and is a computer simulated by software, such as VMware. To reduce resource overhead and improve performance, most blockchains use lightweight virtual machine structures, such as the Ethereum Virtual Machine (EVM). Containers are kernel virtualization technologies that provide lightweight virtualization separating processes and resources. In the Linux operating system, containers are typically created by Docker, which isolates the external environment and provides an independent running environment for smart contracts (Dickson et al., 2018). Hyperledger Fabric employs lightweight Docker containers to execute smart contracts. Docker uses a sandbox mechanism with no interfaces. The program code in Docker runs directly on the underlying operating system, and its execution efficiency is very high.

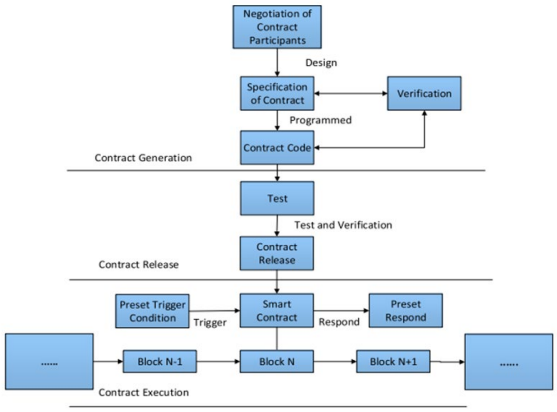

Fig. 4. Smart contract life cycle (Wu et al.,2018)

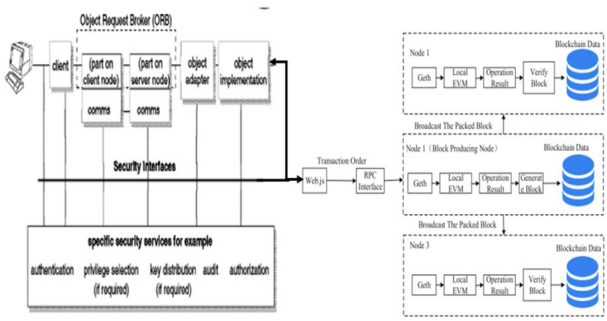

Fig. 5. Object -oriented access control module within a blockchain system on smart contact EVM initiation (Adopted from IBM corporation,2018)

F. Smart Contracts on Blockchains

A smart contract takes the format and feature of computer program code encompassing associated commercial transactions and algorithms (Pagano, 2021). Essentially, it represents the automation of a prearranged contract between involved parties. This unique contractual agreement undergoes automatic verification once preset conditions are triggered, thereafter execution as soon as predefined conditions are met. Smart contracts are not only used in the field of financial transactions, but also include many aspects of social life (Bankykoom et al., 2018). Let the contract drawn up by the transaction parties p1, p2 . . . , pk (k subset of Z+) in the real world be C, the smart contract is recorded as IC, the trusted third party institution is G, and under the supervision of the institution G, the parties to the contract, the result of performing the contract C is recorded as R, that is, R = C(P, G), then R = IC(P), P = {p1, p2, . . . , pk}. Smart contracts automatically complete transaction contracts that require the supervision of a trusted third-party organization in the real world to ensure that the contracts are actually fulfilled. A smart contract outlines a series of commitments in digital form. In the Ethereum system, the smart contract serves as a protocol for controlling digital currency assets using blockchain technology (BitLand, n.d.). In computer terms, a smart contract can be perceived as a segment of code involving interconnected commercial transactions and algorithms. From a public viewpoint, it represents an associated agreement. Upon the fulfillment of preset conditions, the smart

contract undergoes automatic validation and execution (Buterin, et al.,2018)

Smart contracts undergo a life cycle comprised of three stages: contract creation, contract deployment, and contract execution. The creation phase encompasses several steps such as multi-party negotiation, formulation of contract specifications, verification of the contract, and acquisition of the contract code (Wu et al., 2019). Initially, the parties involved in the contract negotiate to clarify each other's rights and obligations, determine the standard contract text and program it, and obtain the standard contract code after verification. The contract generation process involves two important links: the contract specification and contract verification (Cazzola et al.,2018). Contract specifications need to be negotiated and formulated by experts with relevant domain knowledge and contract parties. Contract verification is carried out on a virtual machine based on the system abstract model, which is related to the security of the contract execution process, and the consistency of the contract code and the contract text must be guaranteed. Following the contract generation, subsequent stage is contract release. The signed contracts are disseminated to the nodes in peer-to-peer (P2P) mode, and the node temporarily stores the received contract in memory as it awaits consensus agreement.

The Main Steps of the Consensus Process Include:

(1) Package for contract collection. Each node package temporarily stored contracts in the recent period to form a contract set. (2) Contract blocks are generated and broadcast to the entire network. Calculate the hash value of all contracts in the contract set and then assemble these hash values into a new block and publish it to other nodes in the entire network. (3) Other nodes validate blocks. After receiving the newly broadcasted block, other nodes compare the hash value in the block with the Hash value of the contract set saved for verification. (4) Multiple rounds of comparison, consensus reached, and the entire network broadcast. After several rounds of sending and comparison, all nodes eventually reach a consensus on the newly released contract, and the consensus set of contracts is broadcast to all nodes in the entire network in the form of blocks (Ceglowski et al.,2015).

The code and execution are distributed across multiple nodes, making the process resistant to censorship or interference from a single party (Han et al.,2020). Smart contracts automatically execute when predefined conditions specified in the code are met. This removes the need for a third party to enforce or validate the terms of the contract. In a smart contract, participants are not required to place trust in each other. Trust is instead established through the code and the decentralized consensus mechanism of the blockchain (Guo et al.,2018). This guarantees the integrity of the agreed-upon terms and ensures their reliability. The code and execution of smart contracts are transparent and observable on the blockchain. Participants can verify the contract's status, terms, and outcomes at any time (Guo et al.,2018).

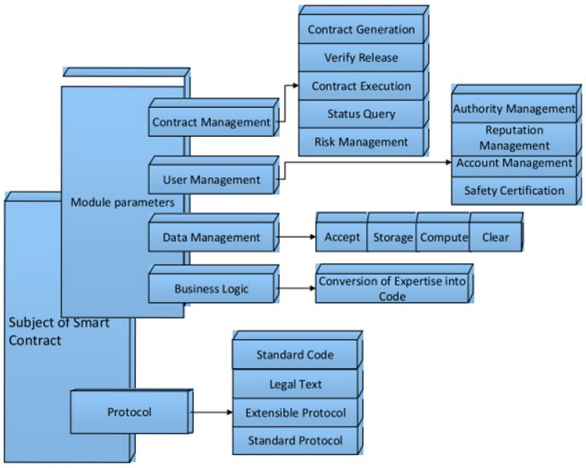

Fig. 6. The main structure of a smart contract (Wu et al.,2019)

Contract execution is based on the ''event trigger'' mechanism once and only if the preset conditions are satisfied. The smart contract subsystem in the blockchain system has transaction processing and preservation functions, as well as a complete state machine for accepting and processing various smart contracts. The smart contract subsystem periodically traverses the state machine and trigger conditions of each contract, and pushes contracts that meet the trigger conditions to the queue to be verified. The contracts to be verified are broadcast to each node. Similar to ordinary blockchain transactions, the node first performs signature verification to ensure contract validity. The verified contract will be successfully executed after reaching a consensus (Chang et al., 2018). The entire contract processing process is automatically completed by the smart contract subsystem built into the bottom layer of the blockchain. In essence, the realization of a smart contract is to give the object digital characteristics: that is, the object is programmed and deployed on the blockchain to become a resource shared by the whole network, and then trigger the automatic generation and execution of the contract through external events, so as to change the state and value of digital objects in the blockchain network. Existing smart contract platforms like Ethereum and Hyperledger feature Turing complete script development languages, expanding blockchain capabilities to support various smart contract applications in finance and social systems (Christidis et al., 2016).

Smart contracts find application across diverse sectors such as financial services (e.g., decentralized finance or DeFi), supply chain management, voting systems, and insurance (Christidis et al., 2016). For instance, in financial smart contracts, terms can delineate conditions for fund transfers, with the contract automatically executing the transfer upon meeting these conditions. Platforms like Ethereum require computational resources for smart contract execution, with participants compensating nodes for processing through "gas" fees. These fees fluctuate based on contract complexity. Smart contracts may integrate external information to initiate actions,

with oracles serving as external data sources to provide real-world data to smart contracts (He et al., 2019).

Compared with traditional contracts, smart contracts have the following inherent significant advantages: 1) Reducing transaction risks. Owing to the immutable nature of blockchain, smart contracts cannot be changed at will once they are released on the chain. Additionally, all transactions recorded and replicated across the distributed blockchain system are fully traceable and auditable. Thus, malicious acts like financial fraud can be greatly mitigated. 2) Reducing administrative and service costs. Blockchain ensures the trustworthiness of the entire system through a distributed consensus mechanism without going through a central broker or intermediary. Once the smart contract stored in any block is triggered, it is broadcast to the entire blockchain network after being verified and executed by the nodes. As a result, administrative and service costs can be significantly reduced by eliminating the need for third-party intervention. 3) Improving the efficiency of business processes. Removing dependency on mediation can significantly improve the efficiency of business processes. For example, once predefined commodity supply chain procedures are met, such as the buyer confirming receipt of the relevant product, financial settlement will be automatically completed in a point-to-point manner, thereby greatly shortening the transaction turnaround time (Chandrasekhar et al., 2018).

1) Challenges of Executing Smart Contracts in Blockchain

While smart contracts offer numerous advantages, they also face several challenges that need to be addressed for broader adoption and improved functionality. Here are some key challenges associated with executing smart contracts on a blockchain. Security is paramount, as vulnerabilities in the code could lead to exploits. Security is a critical challenge for smart contracts. Bugs or vulnerabilities in the code can be exploited during access session, leading to financial losses. Code audits, formal verification, and rigorous testing are essential to mitigate these risks. Auditing and rigorous testing are essential (Guan et al.,2019). Smart contracts often rely on external data sources (oracles) to trigger actions based on real-world events. The reliability and security of oracles are crucial, as inaccurate or manipulated data can lead to incorrect smart contract executions (Gaur et al.,2019).

The traditional blockchain adopts a single-chain data structure, outside the genesis block, where each block has only one predecessor block, and the blocks are serially connected by hash pointers in the sequence of block production time series to form a single chain. In single blockchain system, the smart contracts are executed serially which therefore takes a long waiting time for the contract execution, which results in a very minimal number of contracts executed per second by the system. Moreover, on the distributed single chain structure of the blockchain-smart contract records all the state changes of the blockchain network from its birth to the current moment, and requires each node to maintain a complete data backup. In fact, these massive amounts of contract data continue to grow rapidly. For nodes in the chain, it is extremely difficult to store and synchronize these increasingly large amounts of data, and the contract data storage is difficult to expand (Decker et al.,2017). Blockchain networks, especially those with high transaction volumes like Ethereum, face scalability challenges (Gervais et al., 2016). As the number of transactions and smart contracts increases, the network may experience congestion and slower transaction processing times.

The legal and regulatory status of smart contracts is still evolving. Legal and regulatory challenges may arise as smart contracts operate in a somewhat novel legal and regulatory landscape. Ambiguities in legal frameworks and uncertainties about the enforceability of smart contracts in traditional legal systems can pose challenges, especially in cross-border transactions. The immutability of smart contracts, while a strength in terms of trust, becomes a challenge if there are bugs or vulnerabilities in the deployed code. Once deployed, fixing such issues is difficult, and it requires careful consideration

are transparent, privacy concerns may arise when executing smart contracts that involve sensitive or private information. Solutions such as zero-knowledge proofs are being explored to address these concerns. Updating or upgrading a deployed smart contract is challenging due to its immutability. Developers must consider mechanisms for introducing changes while ensuring backward compatibility and minimizing disruptions (Bangare et al.,2016).

Smart contracts on one blockchain may not be directly compatible or interoperable with those on another blockchain. This lack of standardization can hinder collaboration and limit the potential for integrated applications across different platforms. Developing and understanding smart contracts often requires a deep understanding of blockchain technology and programming languages specific to the platform. Improving the user-friendliness of smart contract development tools is essential for broader adoption (Filippi et al.,2018). The development language of contracts is still immature, and there is a lack of effective detection and processing methods for potential vulnerabilities. For example, the solidity development language of Ethereum lacks safe handling methods for problems such as function variables and operation symbols out of bounds, and most developers do not have enough semantic understanding of these development languages to use Turing machines flexibly, which can easily lead to security vulnerabilities in smart contracts. Since then, Ethereum smart contracts cannot be modified once deployed, and it is particularly difficult to solve the security problem in smart contracts. If the smart contract code design contains errors, there is no direct error-correction method after deployment into the chain. Therefore, it is necessary to design a method for modifying and ending the contract state (Filippi et al.,2018).

A.Conceptual Framework: Proposed Solution

The conceptual framework is a set of broad ideas used to explain the relationship between the independent variables (factors) and the dependent variables (outcomes). The independent variables also known as exploratory variables and

Table 3
Features of Smart contract, challenges and progressive proposed solutions (Odhiambo et al., 2024)

Smart contract programme features	Type of challenge			Proposed solution in progress	Citation
	Performance issues	Privacy Issues	Security issues		
		Content			
Ethereum A smart contract execution framework that supports a multicore architecture, allowing miners and validatots to execute independent conflict free smart contracts in parallel;YODA,ACE,CITA	Inefficient contract execution and difficulty in expanding contract data storage	Trusted data source privacy and contact data privacy disclosure	Smart contracts are vulnerable to potential access security vulnerablities, such as the operating environment, compliation process and program of characterics within smart contract.	Concurrent execution of-chain computing and contact microservices Channel isolation,power limit system can be hard component execution environment Fuzzing testing, symbolic execution, formal verification and other technologies:{SMARTIAN, ETHPLOIT, Oyente, ILF, Solidity, ZEUS}	Wu et al.(2018) Bangare et al.(2016). Gaur et al.(2019) Liu et al.(2018) Filippi et al.(2018) Cheng et al.(2016) Gervais et al.(2016). Guan et al.(2019). He et al.(2019). Christidis et al.(2016). Ceglowski et al.(2015 Cazzola et al.(2018 Buterin et al.(2018 BitLand. (n.d.) Wang et al.(2017) Zhao et al.(2016)
		Threat attack challenge			
Mictract A framework based on microservices BFT-Smat A new two-phase framework based on trusted hardware Intel SGX to improve parallelism between nodes Bitcoin A multi payment channel MPC network using intermediate channel	Low throughput ,data storage difficulties and poor scalability	Artficial steal,control network nodes,profit chain code vulnerability to obtain private information	Cause huge economic losses, leak user information,contract loopholes difficult to repair		

during the development phase. While blockchain transactions

which are presumed cause of changes in the dependent variable which the researcher wishes to explain (Zadeh et al.,2015). A conceptual framework is a diagrammatic representation of how variables interact. It provides a clear concept of the areas in which meaningful relationship are likely to exist (Zadeh et al.,2015). Smart contracts have expanded to include multiple applications and services. It is a dynamic and distributed system which creates several issues that need be taken into accounts when building an access control model that has the element of dynamism and adaptability. Figure 6 presents the conceptual framework of a smart contact with the adaptiveness and dynamism that can be executed on blockchain.

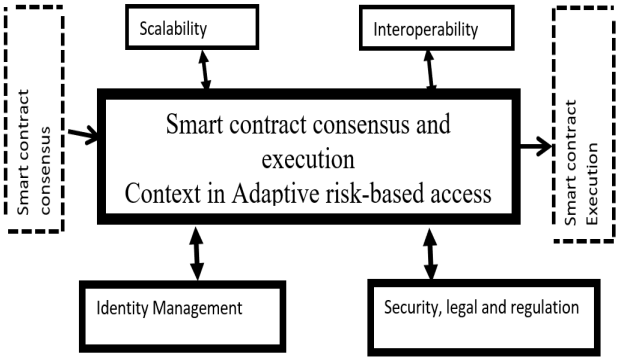

Fig. 6. Conceptual framework for Access control in smart contracts (Odhiambo et al., 2024)

Figure 6 shows scalability, interoperability, security, legal and regulations, and identity management as the independent variable in this study. Smart contract consensus and execution context as the depended variable.

The constraining and intervening variables are also addressed alongside the independent variables in table 4. We intend to investigate these variables on how they relate and interact to determine a secure access and execution of smart contract to provide adaptability and dynamism on blockchain platform as documented herein:

G. Interoperability

Ensuring interoperability is essential for the execution of smart contracts on a blockchain platform, facilitating cross-platform execution, streamlining inter-chain communication, integrating external data sources via systems like oracles, and allowing execution on external applications running on legacy systems. Another crucial aspect influencing an access control model is the formulation of access policies. These policies need to be designed to accommodate multiple users and organizations. While each organization can establish its unique policies, there is a simultaneous need to adhere to the policies set by other organizations.

Interoperability stands out as a pivotal feature in smart contract blockchain technologies. These systems eliminate reliance on centralized third parties for the security of transactions and smart contracts. The decentralized nature of blockchain allows for the distributed recording, storage, and updating of data. As highlighted by Gartner (2018) in the hype circle of emerging technologies, blockchain has the potential to revolutionize data security, enhancing reliability, transparency, and trust. This technology, incorporating elements of cryptography, peer-to-peer networks, and mathematics, effectively addresses synchronization challenges present in traditional distributed databases by merging P2P networking with distributed consensus algorithms. This integration results in reduced costs associated with multi-signature processes and increased transparency. Incorporating the standardized features of the blockchain ecosystem into our model presents a significant opportunity for the execution of smart contracts.

1) Scalability

The smart contract system encompasses billions of devices, generating an extensive volume of data that necessitates substantial processing capabilities. Designing an access model for smart contracts integrated within the blockchain must account for the expanding network size. Scalability, defined as the system's ability to manage growing workloads and accommodate expansion without compromising performance, is a critical consideration (Sharma et al.,2019). To address scalability challenges, various intervening and constraining variables, such as layer solutions, the implementation of sharding techniques involving multiple subsets of data execution portfolios, off-chain computation of data, and optimized consensus dynamic interactions within the smart contract environment, can alter the dynamics of access requirements between users. These adjustments enable the access policies to adapt to diverse situations and changing conditions while making access decisions within the system (Sharma et al.,2019).

The primary objective of smart contracts on the blockchain is to shift trust from a centralized server to the entire system, ensuring security without breaches. This distinctive property proves highly beneficial for enhancing database storage capacity to manage large volumes of data. Validating its intrinsic value constraints, smart contracts can be effectively utilized in financial transactions, as the transaction records are permanently preserved, and unauthorized alterations are prevented unless an intruder gains control of more than 51% accessibility of the network (Bore et al., 2017).

2) Identity Management

Given that a smart contract is self-executing based on the terms outlined in coded agreements, the authentication of the digital identities of involved parties can be achieved through public-private key encryption, biometric authentication, or other digital signature methods. These variables play a crucial role in managing access and delegating authority attributes. In specific access scenarios, there is a need for Ai agents to operate on behalf of users for defined periods. Therefore, an access model should consider the delegation of authority to enhance usability and flexibility. Context awareness, defined by the Cambridge dictionary as the situation within which something happens, is an essential factor when constructing an access control model. Incorporating context awareness enables user interactions, making it imperative to consider real-time contextual information when making access decisions (Bancor et al., 2018).

The existence of ownership uniqueness contributes to identity and accessibility, involving the monitoring of possession based on three main principles: public or permissionless, private or permissioned, and consortium. In a

Table 4
Conceptual framework for access control in smart contracts (Odhiambo et al., 2023)

Independent variable	Intervening and control variable
Identity Management	➢ Presence of uniqueness of ownership
	➢ Monitoring of possession
Security, legal and regulation	➢ Vulnerabilities detection and attacks prevention
	➢ Legal and regulations for anonymization
Smart contract Execution	➢ Processing vast volumes of data
	➢ Extraction of information from massive databases
Interoperability	➢ Fitting the standardized features of ecosystem
	➢ Cost of multi signature and transparent
Scalability	➢ Intrinsic constraints
	➢ Storage capacity
Smart contract consensus	➢ Selection of primary nodes
	➢ Decreasing network broadcasting

public smart contract blockchain, the design aims to eliminate intermediaries from transactions to maintain security, while a private blockchain restricts users from validating actual transactions and creating smart contracts. On the other hand, a consortium smart contract blockchain is partly private and allows specific predetermined nodes to have full control. This presence of ownership uniqueness serves to validate login access to smart contracts (Bancor et al., 2018).

3) Security, Legal and Regulations

Security is a crucial element in a self-executing contract, as once modified and deployed, the contract becomes binding and cannot be altered within the blockchain system. This inflexibility means that vulnerabilities could potentially be maliciously exploited by users once granted access. Granting user access alone is insufficient for the execution of smart contracts; an adaptive access model should be auditable. Therefore, it is essential to collect and store necessary evidence of various access operations. Legal and regulatory considerations become variable factors when publishing a new node on a smart contract and are still evolving within the established legal framework (Aitzhan et al., 2016). This evolving trend emphasizes the need for users to be acquainted with the legal jurisdiction environment in which the smart contract execution takes place especially for our case on freight and logistic business environment.

An access control model for smart contracts, catering to billions of users with diverse security, legal, and regulatory awareness, must provide suitable interfaces to meet the varied needs of users. From the conceptual framework, our intention is to establish a link between the operationalization of the model and the studied dependent variables for the execution of smart contracts. This will be realized through the development of an adaptive model utilizing the fuzzy logic fabric with expert judgment mechanism.

4) Smart Contract Consensus and Execution Context

In a blockchain system that ensures secure access and the execution of smart contracts, it is crucial to establish a validation process through an algorithm such as proof of work or proof of stake. This consensus algorithm ensures agreement among all nodes on the network regarding the nature and state of the contract to be executed. Another essential component is the execution context of the smart contract, where the actual processing of contract terms, such as the transfer of digital assets and the update of records, occurs within lines of code (Gupta et al.,2018).

The blockchain system is open to everyone, allowing anyone to validate and audit transactions. Individuals utilize blockchain technologies to create various applications of their choosing. This type of database exists across different computer systems, forming a peer-to-peer network, eliminating the presence of a single, centralized database or server. However, this decentralized structure tends to increase network broadcasting (Wang et al.,2020). To address this, the selection of primary nodes becomes crucial, initializing the use of digital signatures with public key cryptography.

A key aspect of blockchain technology is the consensus algorithm, determining which users publish the next blocks. When consensus algorithms fail, it leads to issues such as fork problems, dominance issues, and deficient performance of the smart contract blockchain network (Gong et al.,2018). Consensus algorithms should possess properties like safety and consistency, and fault tolerance is affected by the recovery to a pre-selected primary node, addressing the trust problem between nodes. There are two different anonymity sets in a communication system: sender sets and recipient sets (Chen et al.,2019). The adaptive risk-based access control model will also utilize Proof of Bandwidth (PoB) consensus mechanisms to grant access adaptively. In such cases, vast volumes of information need extraction and processing from databases for verification and validation to grant access into the system, involving cryptocurrencies, agreements, documents, or other data.

In the transmission of desired transactions through nodes, a P2P network is involved. Through a recognized algorithm, the node network validates identity and user status. Subsequently, a new block is added to the existing blockchain, containing a hash, verified proof of valid transactions with a timestamp, and the hash of the previous block. This prevents the block from being altered or a block being inserted between two existing blocks (Cheng et al.,2016). Smart contracts executed based on certain conditions can be written into the platform, applicable only to permissioned blockchains with a high level of trust. After solving the proof of work (PoW) puzzle, the block is broadcast to other nodes, detecting vulnerabilities and virtually preventing attacks from intruding. Examples of PoW include Bitcoin, Kovan testnet, and Ethereum. The main goal is to develop a less computational but adaptive risk-based access control model than PoW with better dynamic and robust access security guarantees. The publishing of the new node depends on random waiting time from a secure hardware shell. Regarding legal and regulation for anonymizations, Hyperledger Sawtooth is an example of PoET. In anonymizing the public blockchain with no access restrictions, anyone connected to the internet can participate in reading, writing, or auditing transactions. In a private blockchain, which is permissioned, participants can join only after receiving legal and regulated invitations from network administrators, as seen in a Bankchain type. For a Consortium or Federated smart contract Blockchain, which is semi-decentralized, selected members of the consortium can run the entire node, make transactions, and review or audit transactions, exemplified by Hyperledger consortium smart contract blockchains (Diep et al.,2019).

H. Taxonomy for Secure Identification Mechanism

Throughout this design formulation process, our taxonomy can assist the decision making through enabling a systematic comparison among the capabilities of different design options. The taxonomy also shows the impact of different design options on the quality component attributes. Smart contracts enables

new forms of distributed software architectures, where components can find agreement on their shared states without trusting a central integration point (Gupta et al.,2018).

1) Fuzzy Logic Technique

The fundamental principle underlying fuzzy logic involves the application of a predefined set of rules (if-else statements) in parallel to interpret certain values within the input vector and subsequently assign values to the output vector (Porwal et al., 2015). Fuzzy Logic is based on fuzzy sets in that, unlike classical sets, their membership is not a true-false‘ but not-quite-true-or-false‘ answer (Mathworks, 2021). The figure 2.5a below explains this concept.

A fuzzy set A is stuctured up of ordered pairs and is defined as follows: A={x| mA x) | x Î X} where X is the universe of discourse whose elements are denoted by x and mA(x) is the Fuzzy Membership Function of x in A. This is a value in the unit interval [0,1], where zero[0] means that an attribute has complete non-membership in a fuzzy set; one[1] means that an attribute has complete membership in a fuzzy set, and grades between 0 and 1 mean partial membership in a fuzzy set. This value (grade) is associated with a certain proposition in the domain for the adaptive risk -based access control model. A set of pre-defined rules (if-else-statements) are applied in parallel to interpret some values in the input vector and then assign values to the output vector (Atlam et al.,2021).

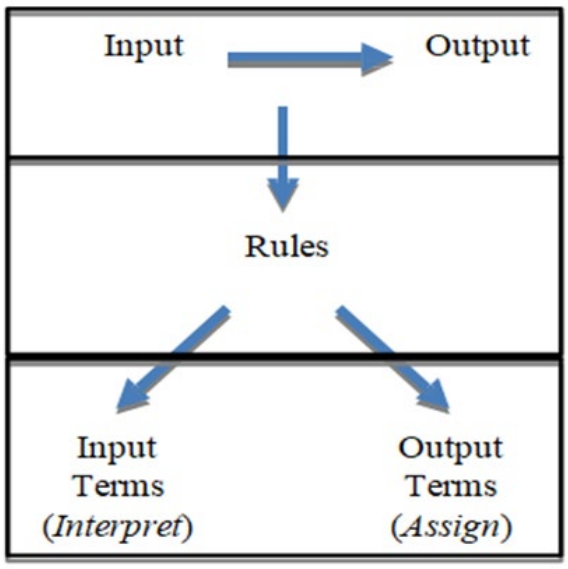

Fig. 7. Fuzzy flowchart (Mathworks,2021)

The classical adaptive access control model relies on Boolean or crisp sets, where membership in a set is determined by a characteristic function that assigns a value of either 1 (true) or 0 (false) to each individual in the universal set X. In contrast, the proposed adaptive risk-based access control model employs dynamic continuous and fuzzified sets. (Atlam et al., 2021; Bai et al.,2016).

A Fuzzy Membership Function (FMF) is a curve defining how each point in the input space is mapped to a membership value between 0 and 1. The choice of FMF depends on the application domain, considering factors like simplicity, convenience, speed, and efficiency. FMFs can be based on functions such as piecewise linear, Gaussian distribution, sigmoid curve, quadratic and cubic polynomial curves. Gaussian and sigmoidal functions, which are S-shaped and open to the right, are suitable for modeling access control in adaptive risk-based models. They are proven to be appropriate for linguistic variables and are supported in the MATLAB Fuzzy Logic Toolbox (Mathworks, 2021).

There are three stages in fuzzy modelling: fuzzification of inputs, logical inference procedures, and defuzzification as shown in figure 2.6. Fuzzification involves generating FMFs for inputs to represent degrees of membership between 0 and 1. Logical inference procedures combine fuzzy sets into a synthesized fuzzy set, resolving the antecedent (IF x) part(s) to a single number between 0 and 1. Defuzzification transforms the synthesized fuzzy set back to crisp sets, assigning an entire fuzzy set to the output based on the consequence of a fuzzy rule. In real-life scenarios, multiple fuzzy rules are used, and the output of each rule is aggregated into a single output fuzzy set, which is then defuzzified into a single number through the fuzzy inference mapping process (Zadeh et al., 2015).

Fig. 8. Fuzzy Modelling Stages (Carranza et al., 2015)

In Fuzzy Logic, traditional logical operations have been adapted to work with degrees of truth. The standard logical operations are modified as follows:

Fuzzy intersection or conjunction (A AND B): min(A, B)

Fuzzy union or disjunction (A OR B): max(A, B)

Fuzzy complement (NOT A): 1 - A

These operations retain the values of the standard logical operations truth table. IF-THEN rules in Fuzzy Logic construct complete sentences, following the format: IF x is A THEN y is B, where A and B are linguistic values defined by fuzzy sets on the ranges (universes of discourse) X and Y, in that order. The IF-part (x is A) is called the antecedent or premise, while the THEN-part (y is B) is called the consequent or conclusion. For instance, IF cyber threat level is high, THEN the user will be denied OR Else have limited access control to execute the smart contract (Carranza et al., 2015).

The proposed adaptive risk-based access control model incorporates fuzzy input data sets specific to security access, allowing for a more nuanced representation of security factors. Some of these sets will include:

Access Control:

User Permissions: "full access," "limited access," "no access"

Authentication Strength: "strong authentication," "moderate authentication," "weak authentication"

Blockchain Network Security:

Node Reputation: "trusted nodes," "partially trusted nodes," "untrusted nodes"

Network Latency: "low latency," "moderate latency," "high latency"

Smart Contract Vulnerabilities:

Vulnerability Severity: "critical," "moderate," "low"

Patch Management: "timely patches," "occasional patches," "no patches"

Transaction Security:

Transaction Anonymity: "fully anonymous," "partially anonymous," "non-anonymous"

Transaction Confirmation Time: "fast confirmation," "typical confirmation," "slow confirmation"

External Threats:

Cyber Threat Level: "low threat," "moderate threat," "high threat"

Malware Detection: "effective detection," "partial detection," "ineffective detection"

These fuzzy input data sets, employing fuzzy logic, introduce a level of granularity and flexibility in assessing and managing security risks within smart contracts in a blockchain system (Zadeh et al., 2015). They allow the model to handle uncertainty and imprecision, leading to more informed decisions that enhance the security of smart contracts.

2) Expert Judgment Thematic Mechanism

Expert judgment is a potent tool in risk analysis, offering diverse solutions and decisions across various domains, including psychology, criminal justice, financial forecasting, political science, and decision analysis, where expert judgment stands as the primary source of valuable information. When practical data is insufficient to describe the probability and impact of a specific incident, expert judgment becomes a valuable approach, providing a subjective evaluation based on the expert's experience and insights gained through careful group focus interviews. Estimating the probability of an incident in a risk analysis, especially for rare and extreme events, is a challenging task, given the inherent uncertainty (Walters et al., 2021). Expert judgment involves expressing inferential opinions derived from knowledge and experience (Yin et al., 2016). It is frequently employed to assess uncertain parameters in a probabilistic manner and evaluate various components of a model.

This complexity is particularly evident when attempting to assess the security risks associated with access control operations for smart contracts, as illustrated in Figure 9, which presents a thematic taxonomy of expert judgment in smart contract access control mechanisms.

In Figure 10, a hash function is depicted, capable of taking input data of varying lengths and producing a distinct fixed-length message as output. If any alterations are made to the input, the resulting output becomes entirely different (Zhang et al., 2015). In the adaptive risk-based access control model, hash functions play a pervasive role in managing user identification. Each block containing data undergoes hashing, fuzzification and modifications which will result in output sets of varying sizes within the MATLAB black-box toolkit.

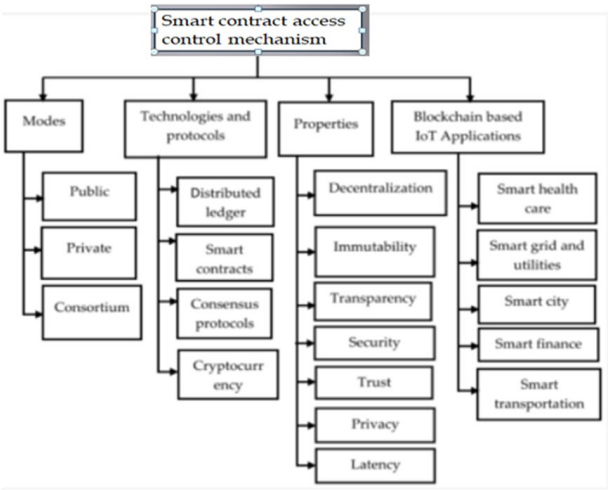

Fig. 9. Expert judgment thematic taxonomy of smart contract access control mechanism (Wang et al., 2017)

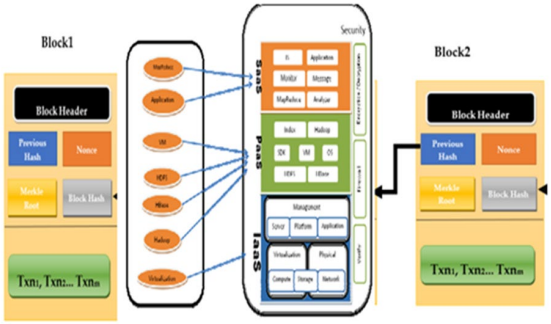

Fig. 10. Architecture matrix for smart contract access model on blockchain application (Zhang et al.,2015)

The figure presented in our study illustrates a scenario where a user attempts to access a session. During the initialization of a smart contract transaction, data stored in a block is modified by incorporating a Merkle Tree or hash tree. Each node in this Merkle tree is represented as a leaf and labelled with a block for user validation. This Merkle tree facilitates secure and efficient storage of large data structures once the user is granted permission. The inclusion of a timestamp element enables the tracking of the creation or modification time of smart contract documentation in a secure manner (Reyna et al.,2018). The nonce, a 4-byte value that starts with 0 and increments with each hash calculation, is distributed within the adaptive model whenever a hash is solved and performed (Chandrasekhar et al., 2022).

Figure 10, shows identity management test for a user during access session, where Blocks 1 and the output block (Block 2) encompass applications that dynamically vary across multiple

transactional dimensions such as size, interoperability, data, users, dependencies, and technologies. This diversity manifests as the applications operate across different phases and interact with various network platforms in a freight and logistics business environment.

I. *Summary of Literature*

Smart contracts have captured the attention of experts, specialists, and researchers in both academia and industry due to their potential to revolutionize daily life activities (Ruddick et al., 2016). While their utilization brings numerous benefits, it also introduces various security challenges. The current access control models, characterized by rigid and static structures with predefined rules yielding consistent results across different situations, fall short in providing the necessary security for such execution-rendering systems. In response, this study presents an adaptive and dynamic risk-based access control model. The proposed model leverages real-time and contextual information from smart contracts associated with access requests to autonomously determine access decisions.

User attributes collected during access requests, data sensitivity, action severity, and user risk history serve as inputs to estimate the risk value for each access request in the proposed model. To enhance abnormality detection capabilities, smart contracts monitor user activities throughout the access session, detecting and preventing malicious attacks from authorized users. Recognizing the pivotal role of selecting an optimal risk estimation technique in building a risk-based model, we discuss common risk estimation taxonomies used in related models. Future work aims to conduct interviews with security experts to determine the ranges of fuzzy sets and fuzzy rules required for implementing the secure identification risk estimation process.

This chapter encompasses the theoretical framework, a critique of literature-research gaps related to the adaptability of existing models. A conceptual framework is then developed as it seeks to explore the design and creation of the proposed adaptive model from an architectural perspective.

3. Research Methodology

A. *Introduction*

The methodology chapter discusses the techniques that will be used to solve the research problem. The chapter entails Section 3.2 which explains the research philosophy, section 3.3 explains the research design, while section 3.4 data collection and preparation methods. Section 3.5 outlines the proposed architectural design for the model, whereas section 3.6 explains evaluation of the suggested model as outlined in the conceptual framework. Ethical consideration is discussed in section 3.7.

B. *Research Philosophy*

Philosophical concepts are closely linked to specific research designs, influencing the selection of a suitable research design approach for a study. In this research, we adhere to the philosophical construct of scientism, which asserts the idea that the scientific method and approach are universally applicable. Our research is grounded in the scientific methods, enabling us to explore various datasets, manipulate them, and deconstruct them to test fuzzy datasets using the MATLAB toolkit—a tool that fundamentally relies on scientific principles.

C. *Research Design*

The research adopts a Mixed-Method research design which includes Experimental research design that will be used to design and develop the model while Action research design will be used to test and evaluate the model. Therefore, we consider a Mixed-methods research design that integrates qualitative and quantitative data collection and analysis techniques.

In Experimental research design we shall conduct experiments where different risk levels are simulated, and the response of the adaptive model is observed. Quasi-experimental designs will be inter-looped in-order to manipulate variables and measurement of outcomes. The performance model will be experimented against traditional access models.

Action research design that will be employed in the context of testing the model will involve iterative cycles of design, testing, evaluation, and refinement based on feedback from stakeholders. Table 5 shows the input risk assessment factors that will provide action feedback.

1) *The Experimental Model Set-Up*

Experimental research design will be used to design and develop the model by manipulating variables in a controlled environment. By employing experimental research design, conducted within a controlled environment, this then will provide empirical evidence of access models' effectiveness, usability, and performance under varying risk conditions. This approach will help in identifying strengths, weaknesses, and areas for improvement. Experimental setup procedures will involve the design and configuration of a test environment that simulates real-world scenarios and conditions relevant to the evaluation objectives. We shall use the risk assessment factors identified in table 3.1 for testing, access control policies, and adaptation mechanisms to be tested and evaluated.

The experimental setup, illustrated in Figure 3.1, will involve the iterative execution in three phases 1) Pilot, 2) Exploratory, and 3) Confirmatory Experiments (PiECEs). Stakeholders, including representatives from freight business enterprises, law firms, and technology hubs, will be actively engaged in the evaluation process during each cycle of PiECEs. This participatory approach ensures continuous feedback and refinement of the solution.

1) Pilot Experiments: Pilot experiments will provide preliminary information. Pilot experiments will be conducted to gain insights into the behavior of various elements and components. It also allows for a systematic and participatory investigation, ensuring that the adaptive model is refined and validated through multiple cycles of experimentation and stakeholder evaluation.

2) Exploratory Experiments will be employed to investigate response patterns to parameter variations or interventions within the adaptive risk-based access control model. It aims to

Table 5
Risk assessment factors, access control policies, and adaptation mechanisms to be tested and evaluated in controlled environment (Odhiambo et al., 2024)

Parameters	Risk conditions	Variables-adaptation mechanism
Variable Manipulation	Risk thresholds	Adjusting the thresholds for defining different risk levels to assess how the model responds to varying levels of risk.
		Modifying the access control policies based on risk assessment results to evaluate their impact on security and usability.
	Access control policies	Testing different algorithms or strategies for dynamically adapting access control decisions based on changing risk conditions.
	Adaptation mechanisms:	
Controlled Environment	Control extraneous variables that may influence the outcomes to isolate the effects of the manipulated variables on the adaptive model's performance.	Set up a simulated network environment to precisely control risk factors, user behaviors, and system configurations to evaluate the adaptive model under various conditions.
Random Assignment	To ensure the validity of experimental results, participants or subjects should be randomly assigned to different experimental conditions or treatment groups.	System administrators or users randomly assigned to groups using the adaptive model versus traditional access control models. This helps minimize bias and ensures that any observed differences in outcomes are attributable to the treatment.
Measurement of Outcomes	Involves measuring outcomes or dependent variables to assess the impact of the manipulated variables.	
	Security effectiveness	The number of security incidents, unauthorized access attempts, or breaches detected under different experimental conditions.
	User satisfaction	Perceptions of system administrators and users regarding the usability, effectiveness, and overall satisfaction with the adaptive model.
	System performance	Metrics such as response time, throughput, and resource utilization to evaluate the impact of the adaptive model on system performance.
Statistical Analysis	Experimental data are typically analyzed using statistical techniques to determine the significance of observed differences between experimental conditions.	Statistical tests such as t-tests, ANOVA, or regression analysis will be used to compare outcomes across different experimental conditions and identify factors that significantly influence the performance of the adaptive model.

generate hypotheses for subsequent formal testing in confirmatory experiments. Confirmatory Experiments will then be conducted to rigorously test and validate the sample data sets confirming the hypotheses established prior to the initiation of all experiments.

The Evaluation of stakeholders, will be implemented immediately after the completion of the Pilot and Exploratory Experiments. Path 4b allows a return to the regular PiECEs cycle, ensuring the resumption of Exploratory Experiments interrupted by path 2b.

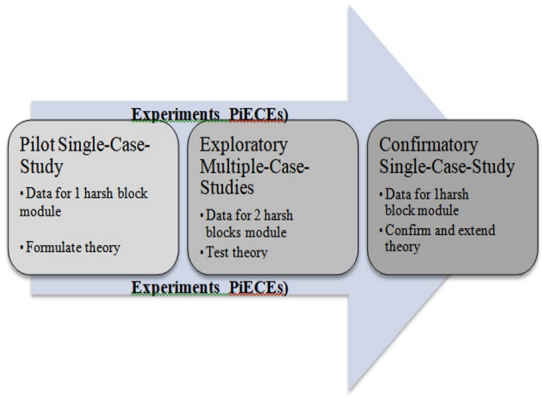

Fig. 11. Pilot Experimental design (Franklin et al., 2012)

The exploratory experimentation will be recursively carried out until all issues are resolved. In developing all the various system modules exploratory experiments will involve:

1) The design of computer programs (code that implements with the fuzzy logic set modules) using fuzzy sets in the MATLAB toolkit;
2) System development
3) User (Evaluation by Stakeholders-freight business enteprises) Testing;
4) Model evaluation to adjust various variables in the system program code.

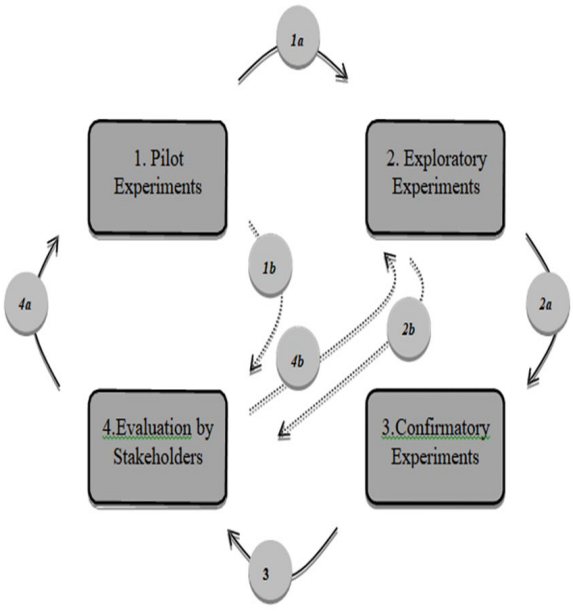

Fig. 12. The phases of PiECEs Experiments Model (Franklin et al., 2012)

D. Data Collection and Preparation Methods

In order to design, develop and test the proposed model. researcher shall then collect data ensuring that data collection procedures are consistent, reliable, and aligned with these research objectives. Quantitative data will be collected through experiments and simulations, while qualitative data will be gathered through focus groups (for expert judgment), and observations (output data).

The establishment of an effective security system, which dynamically adapts access permissions based on evolving risks and user behaviors, relies significantly on the meticulous collection and preparation of data. Some of the methods that will be employed in collection of data from diverse sources, includes the establishment of data pipelines, integration with Security Information and Event Management (SIEM) systems, and the utilization of log aggregators. The procedures therefore will entail;

1) *Coding:* The process of assigning codes to specific variables or elements within the data to facilitate organized analysis and interpretation. For the coding process, sub-codes will be derived from research objectives, questions, the research context, theoretical constructs, and the conceptual framework. The coding scheme remains flexible to accommodate emerging sub-codes from input data, and operational definitions will be updated accordingly. The codebook will guide the first-order coding, employing a descriptive coding technique. To address construct validity, multiple data sources, including reports, will be utilized to ensure converging findings. Reliability will be maintained through programmatically retrieving and storing analyzed transactional stages locally, maintaining a qualitative codebook of codes, and developing matrices from labelled data blocks.
2) Observation of Computed Output: Actively observing and analyzing the computed output generated by the adaptive risk-based access control model.
3) smart contracts, documentation, harsh blocks, and

Table 6
Data source and mining techniques assigned to projected mined data sets(Odhiambo et al., 2024)

Data mining Technique{either Ai-Machine-Deep learning}	**Examples**	**Data set assigned for this technique**
Classification;(Ai) categorizing data into predefined classes or categories based on attributes or features	Decision trees, support vector machines (SVM), naive Bayes classifiers, and k-nearest neighbors (KNN) algorithms	Authentication logs, Security incident reports, User profiles and attributes, Resource attributes, Historical access data, Security event logs and Threat intelligence feeds.
Clustering;(Machine learning) ensembles similar data points together based on their inherent characteristics.	K-means clustering, hierarchical clustering, and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering algorithms.	
Association Rule Learning;(Ai) Picks up interesting relationships or associations between variables in large datasets.	Apriori and FP-growth algorithms used for association rule mining	
Regression analysis predicts continuous numerical values based on input variables[machine learning]	Linear regression, polynomial regression, and logistic regression techniques.	
Anomaly Detection also known as outlier detection, identifies data points that deviate significantly from the norm or expected behaviour, statistical methods, clustering-based approaches, (machine learning) algorithms	Isolation forest and one-class SVM used for anomaly detection.	
Text mining involves extracting valuable insights and patterns from unstructured text data. (Ai)	Techniques such as natural language processing (NLP), sentiment analysis, topic modeling (e.g., Latent Dirichlet Allocation), and named entity recognition (NER) used in text mining	Authentication logs, Security incident reports, User profiles and attributes, Resource attributes, Historical access data, Security event logs and Threat intelligence feeds.
Time series analysis deals with analyzing data collected over time to uncover patterns, trends, and seasonal variations. (deep learning)	Methods such as autoregressive integrated moving average (ARIMA), exponential smoothing, and Fourier transforms are applied in time series analysis.	
Dimensionality reduction techniques aim to reduce the number of features or variables in a dataset while preserving its essential information. (Deep learning)	Principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), and linear discriminant analysis (LDA) common dimensionality reduction techniques.	Authentication logs, Security incident reports, User profiles and attributes, Resource attributes, Historical access data, Security event logs and Threat intelligence feeds.
Ensemble methods(Ai) combine multiple models to improve predictive performance and reduce overfitting.	Techniques like bagging (e.g., random forests), boosting (e.g., AdaBoost, gradient boosting), and stacking are popular ensemble methods used in data mining.	
Deep learning techniques, powerful tools for data mining tasks, particularly in handling complex data types like images, speech, and sequential data.	artificial neural networks (ANNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs),	

Sources of Data sets	Valuable data sets to be mined from sources
i)User Behavior • Public Repositories: GitHub repositories of blockchain projects might have datasets related to user interactions and security logs. • Academic Databases: Research papers and datasets published by universities and research institutions. ii) Security Data • Public Repositories: GitHub repositories of blockchain projects might have datasets related to user interactions and security logs. • Academic Databases: Research papers and datasets published by universities and research institutions. iii)Public Blockchain Data: • Etherscan (Ethereum blockchain explorer): Provides detailed data on Ethereum transactions and smart contract interactions. • Blockchain.com: Offers transaction data for Bitcoin and other major cryptocurrencies. • The Graph: A decentralized protocol for indexing and querying blockchain data, useful for retrieving specific smart contract interactions. iv)Network Data: • Network Monitoring Tools: Tools like Wireshark or Splunk to be used to collect data on network conditions and server performance. • Cloud Service Providers: AWS, Google Cloud, and Azure provide logs and metrics that can be analyzed for network and environmental conditions. v)External environment and legal Data: • Cryptocurrency Market Data: Websites like CoinMarketCap and CoinGecko provide historical price data and market trends. • Social Media APIs: Twitter API, Reddit API for sentiment analysis. • Regulatory Data: Websites of regulatory bodies like the SEC, FATF for compliance requirements.	i)User Behavior Data: • User profiles and historical activity logs. • Behavioral biometrics (e.g., typing patterns, mouse movements). • Geolocation data. • Login patterns and access times. ii)Security Incident Data: • Historical data on security breaches and fraud attempts. • Known malicious addresses and entities. • Types of attacks encountered (e.g., phishing, malware, DoS attacks). • Response times and outcomes of past incidents. iii)Blockchain Transaction Data: • Details of blockchain transactions (e.g., sender, receiver, amount, timestamp). • Smart contract interaction logs. • Frequency and volume of transactions per user. • Patterns of successful and failed transactions. iv)Network Data: • Network conditions (e.g., latency, bandwidth). • Server load and performance metrics. • System uptime and downtime records. v)External environment and legal Data: • Market trends and cryptocurrency price fluctuations. • Regulatory changes and compliance requirements. • News articles and social media sentiment analysis related to blockchain and cryptocurrency.

board.

4) *MATLab Simulations:* Leveraging MATLab simulations to model and simulate various scenarios within the adaptive model for experimental analysis.

1) Data Sources

Data mining techniques will be transacted through Artificial intelligence (Ai) route to retrieve seven (7) data sets which include Authentication logs, Security incident reports, User profiles and attributes, Resource attributes, Historical access data, Security event logs and Threat intelligence feeds. This will involve the use of methodologies and algorithms to extract meaningful patterns, trends, and insights from large datasets using the common data mining techniques which include:

These techniques will be applied individually or in combination, depending on the nature of the dataset and the specific objectives of the data mining task. These techniques will be employed to uncover hidden patterns, relationships, and knowledge from structured, semi-structured, and unstructured data for our model. The target population for the study consist of about 350 key data server repositories/sources.

The Data to be Collected as Presented in Table 6 Will Include:

Authentication logs, Security incident reports, User profiles and attributes, Resource attributes, Historical access data, Security event logs and Threat intelligence feeds using data mining technique that links up with the variables as was presented in the conceptual framework.

The table 7 shows a caption of such repositories as captured with the presumed output for each data set. Data sets will be clustered into scalability, interoperability, security and Identity management sources out from which the relevant input data type will be collected from the sample size.

Table 7
Relevant input data types from the mined data sets (Odhiambo et al., 2024)

Independent variable input	**Scalability;**	**Interoperability;**	**Identity Management;**	**Security and regulation;**
	➢ Resource attributes,	➢ Historical access data,	➢ User profiles and attributes ➢ Authentication logs	➢ Incident reports ➢ Threat intelligence feeds
Intervening Variable	Smart Contract Vulnerabilities ➢ External Threats ➢ Third-Party Services ➢ Contractual Terms	Contract Design and Code ➢ Third-Party Services ➢ Transaction History	Access Control ➢ Data Encryption	Blockchain Network Security: ➢ Transaction Security ➢ Incident Response ➢ Regulatory Compliance ➢ Risk Tolerance

2) The Cleaning Preprocessing and Annotation of the Sample Size Data Sets

During this phase, meticulous data preparation becomes paramount. The data cleaning process involves actively

addressing various tasks to ensure the quality and integrity of the data. This encompasses tasks such as handling missing data through imputation or removal, eliminating duplicates to maintain data consistency, and conducting necessary data transformations. Transformations include annotating date/time conversions and applying one-hot encoding for categorical variables. The annotation of sampled data sets will significantly contribute to risk assessment, incorporating factors such as user access frequency, behavior patterns, and resource sensitivity scores.

Furthermore, data labels will be assigned, and risk levels defined as low, medium, or high. Historical data will be semi-automated labelled accordingly, facilitating model training. To uphold privacy and adhere to data protection regulations, sensitive data, such as user personally identifiable information, will either be anonymized or encrypted. Numerical features will undergo scaling or transformation to ensure uniformity across features, ensuring compatibility with the adaptive risk-based model.

In order to ensure high-quality data for accurate risk assessment within the adaptive risk-based access control model, attention will be devoted to addressing issues such as missing values, duplicates, and inconsistencies.

3) The Splitting of the Data into Training [75%] Testing [15%] and Validation [10%] Sets

The datasets will undergo division into training sets (75%), validation sets (15%), and test sets (10%) to facilitate comprehensive model development, evaluation, and validation. If an imbalance is detected within the dataset, techniques such as oversampling or under sampling will be applied to ensure a balanced representation of risk levels throughout the validation and test phases.

The selection of the appropriate Membership Functions (MFs) relies on the available dataset. The comparison of results between training data and real data, along with the calculation of error values using Mean Average Percentage Error (MAPE), will guide the selection process. Various MF techniques, including trapezoidal, Gaussian, triangular, sigmoidal, and bell-shaped waveforms, will be employed. Triangular MF, efficient in representing expert knowledge and streamlining the calculation process, will be used to depict input and output fuzzy sets in the proposed adaptive risk-based access control model.

The testing phase involving 15% of the sample datasets will define criteria for how output risk changes concerning input risk factors. This will be achieved through fuzzy rules acting as the knowledge base of the fuzzy logic system, utilizing IF-THEN statements to describe actions or outputs based on specific input combinations. The accuracy and efficiency of fuzzy rules will be ensured by considering different risk factors and their combined behavior in producing output risk through a machine learning algorithm.

In the validation of the remaining 10% of datasets, security experts will contribute by providing appropriate fuzzy rules based on their knowledge and experience. During testing, defuzzification will be employed to convert fuzzy variables into crisp variables. This process will involve using defuzzification methods such as mean of maximum, center of area (centroid), modified center of area, height method, center of sum, and center of maximum. These tests aim to ensure data accuracy and evaluate the performance of the adaptive risk-based access control model.

4) Data Manipulation Using the MATLAB Fuzzy Logic Toolkit

The MATLAB Fuzzy Logic Toolkit will be applied to model freight &logistics business knowledge for secure identification management and the execution of smart contracts within the proposed model. Fuzzification of inputs involves determining the degree to which they belong to appropriate fuzzy sets through membership functions, converting classical logic into fuzzy linguistic variables. In this stage, risk factors are transformed into linguistic variables, making them easily understandable. Three fuzzy sets, namely Low, Moderate, and High, will represent action severity, user context, and risk history. Resource sensitivity will be represented by Not Sensitive, Sensitive, and Highly Sensitive fuzzy sets. For the output, five fuzzy sets—Negligible, Low, Moderate, High, and Unacceptable High—will be employed.

By employing the fuzzy logic technique, subjectivity can be reduced to an acceptable level. Quantitative input data allow subjectivity to shift to the rule creation process, providing better control. While subjectivity cannot be entirely eliminated, expert judgment becomes a significant information source in decision-making operations, especially in risk analysis during the smart contract execution phase. Correct numerical data describing incident frequencies and their impact are often unavailable in most risk-based models (Ruddick et al.,2018). In cases where quantifying risk value using classical approaches is complicated, expert judgment can offer a correct risk value for a specific scenario, particularly when appropriate experts are involved.

Expert judgment will be sought through focus-group discussions with individuals possessing deep knowledge and expertise in system security to test the risk estimation process. In summary, the five stages will be employed to implement the fuzzy logic system with expert judgment for estimating security risks in the proposed adaptive risk-based access control model using the MATLAB Fuzzy Logic Toolkit.

5) Processing of Data Sets Using the MATLAB Fuzzy Logic Design Technique:

The inputs clustered as shown in table 3.3 will employ a dynamic continuum of fuzzified sets, exemplified by the User Permission Index (UPI) classes with weightings as follows: Fuzzified User Permission - UPI Full Access $\leq$ -2.0, UPI Limited -2.0 < UPI $\leq$ -1.5, and UPI No Access -1.5 < UPI $\leq$ -1.0. In this framework, the characteristic function for full access permission implies that UPI values of -2.00 and -1.49 are not in the "Full Access" set, while values -1.99 and -1.50 are within this set and also fall within the limited access range. Essentially, all these values signify full access but with differing degrees of

weight. Fuzzy Logic becomes instrumental in allowing membership in more than one set; for instance, -2.00 can belong to both "Full Access" and "Limited Access" sets, while -1.49 can simultaneously be in the "Limited Access" and "No Access" sets, each at varying degrees.

The membership of a user's permission level in the "Grant Access" set is not a binary true-false mapping but instead constitutes a continuous range of values ranging between "false" and "true."

Fuzzy Membership Functions (FMFs) are commonly defined using standard basic functions like piecewise linear functions, Gaussian distribution functions, sigmoid curves, quadratic and cubic polynomial curves. Gaussian and sigmoidal functions, two S-shaped membership mirror-image functions opening to the right based on polynomial curves, are particularly suitable for modelling (Stable et al.,2014). These curves offer the advantage of smoothness and non-zero values at all points, and they are supported within the MATLAB Fuzzy Logic Toolbox, which will be utilized for modelling data sets in the proposed model.

A set of predefined rules (if-else statements) will be concurrently applied to interpret certain values in the input vector and then assign values to the output vector. Figure 12 illustrates how, for example, the input data sets will undergo the process of crisp-fuzzification and be output as defuzzified sets in the formulated adaptive risk-based access control model.

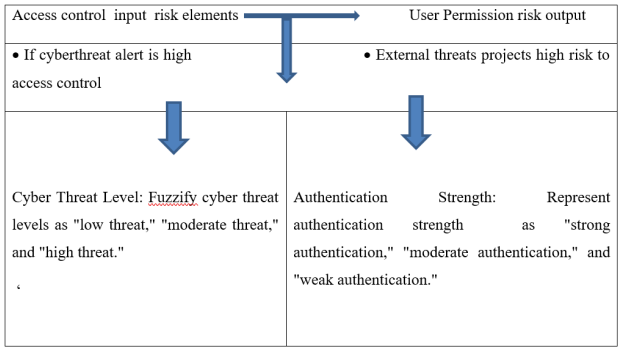

Fig. 13. Fuzzy flowchart with input and output vectors for adaptive risk-based access model(Odhiambo et al., 2024)

Fuzzy Modelling Design Stages with Logical Operations:

Given that in Fuzzy Logic, the truth of any statement is a matter of degree, the standard logical operations will be modified to work for in Fuzzy Logic within our adaptive risk based access model as follows: •Fuzzy intersection or conjunction; A AND B <> min(A,B) •Fuzzy union or disjunction; A OR B <> max(A,B) •Fuzzy complement (NOT A) <> 1-A.

The operations above maintain the values of the standard logical operations truth table.

IF-THEN rules are used in Fuzzy Logic to construct complete sentences; they are formatted as: IF x is A THEN y is B; where A and B are linguistic values defined by fuzzy sets on the ranges (universes of discourse) X and Y, respectively. The IF-part of the rule x is A is called the antecedent or premise, while the THEN-part of the rule y is B is called the consequent or conclusion. For instance, IF cyber threat levels-1 high, THEN the user will be denied OR Else have limited access control to execute the smart contract.

E. Empirical Framework for the Design of Adaptive Risk - Based Access Control Model

Empirical research involves experimentation, observation, and the measurement of phenomena, relying on the first-hand experiences of the researcher. Yaga (2018) asserts that although data collected in such research may be compared against theories or hypotheses, the results are ultimately rooted in real-life experiences. In the context of this study, we introduce an adaptive risk-based access control model and apply it to the empirical execution of a smart contract—from consensus to execution—while addressing key security and legal aspects of identification management and user validation during execution sessions, as outlined in the conceptual framework within the block chain platform.

To safeguard system resources by restricting access solely to authorized users, access control models are categorized into classical and dynamic approaches. Classical models lack adaptability to changing system conditions, relying on predefined rules that yield the same outcomes in different situations. Conversely, dynamic access control models leverage access rules, real-time information, and contextual data to make access decisions. An innovative approach to data protection and information sharing is the adaptive risk-based access control model, which employs security risk as a criterion for access decisions throughout the users active session. This model conducts a risk analysis to evaluate the risk associated with every users' access request.

Security risk, is potential harm arising from existing or imminent operations, serves as the foundation of the adaptive risk-based model with a flexible access control nature that considers environmental contextual information gathered during access requests and addresses exceptional access requests (Atlam et al.,2021). The proposed model should prove efficient in handling unexpected situations that may lead to policy violations due to imperfect policies. To implement this, we propose the development of an adaptive risk-based access control model, module 1,2,3 and full module as illustrated in Figures.3.4. The model utilizes four inputs—user context, resource sensitivity, action severity, and risk history running on block chain platform while executing smart contracts. These inputs inform the risk estimation module, responsible for assessing the overall risk value tied to each access request. Subsequently, the estimated risk value is compared with predefined risk policies to determine whether to grant or deny access. To enhance abnormality detection, we suggest incorporating smart contracts to track and monitor user activities during access sessions, mitigating potential malicious attacks and preventing sensitive information disclosure.

The proposed model incorporates real-time user context

features, considering environmental attributes related to the user during access requests, such as location and time. Resource/data sensitivity defines the importance of data susceptible to improper access, a subjective determination necessitating security experts' input for effective classification. Different types of data carry varying sensitivity levels, each assigned a sensitivity metric within the freight and logistics smart contract. When users specify actions on a resource, action severity gauges the impact on system resources, with security experts through expert judgment categorizing actions and assigning severity metrics. This results in a risk metric associated with each action on a specific resource. User risk history captures past risk values for various user actions, aiding in distinguishing between good and malicious users. The risk estimation module, a crucial component, uses input risk factors to measure the risk value associated with each access request. The ultimate goal is to develop an efficient risk estimation method utilizing real-time information to precisely control access operations.

The estimated risk value undergoes comparison with risk policies to determine access decisions. Risk policies establish access boundaries and conditions for granting or denying access, defining a threshold value. If the risk value falls below this threshold, access is granted; otherwise, access is denied. The process flow of the proposed risk-based model, as illustrated in figure. 13, and 14 begins with a user sending an access request to the access control manager, specifying the resource or data to be accessed and the intended action. The access control manager then collects contextual information related to the user, including location and time, the sensitivity level of the resource, and the severity of the specified action, alongside the user's previous risk history records.

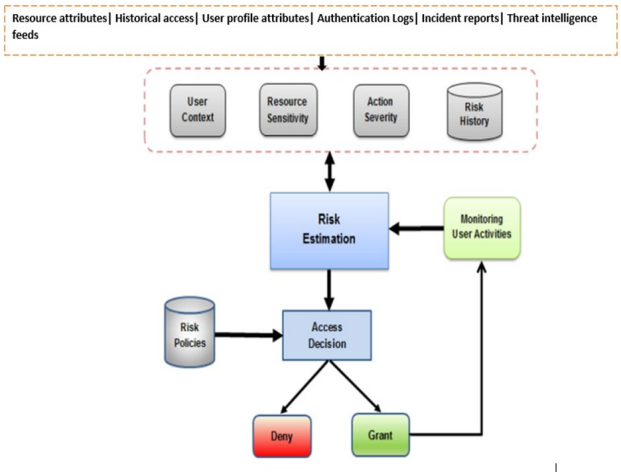

Fig. 14. Proposed adaptive risk-based access control architecture model-module 1(Odhiambo et al.,2024)

The process flow of applying smart contracts to monitor user activities during access sessions is shown in figure. 14.

The risk estimation module utilizes gathered information to assess the risk associated with the requesting user, and subsequently compares the measured risk value with predefined risk policies to make access decisions. If the risk value falls below the threshold specified in these policies, access is granted; otherwise, access is denied. This process presents two scenarios. In the first scenario, where access is granted, smart contracts play a pivotal role in tracking and monitoring user activities during the access session. The smart contract ensures adherence to contract terms and conditions, issuing warnings and terminating the session if any malicious activity is detected. The second scenario involves denying access, prompting the system to request additional proof of identification from the user to reduce false-positive rates. Correct credentials result in access being granted, with subsequent monitoring, while incorrect credentials lead to access denial. Classical access control models lack the ability to detect malicious actions and safeguard system resources after access is granted.

The proposed model enhances system flexibility and abnormality detection capabilities by employing smart contracts to monitor user activities during access sessions. The risk estimation module dynamically adjusts user permissions based on behavior in access sessions, reducing privileges or terminating access sessions upon detecting abnormal actions. Smart contracts, known for their flexibility, can securely encrypt and store data, restrict access, and execute logical workflows. Implementing a smart contract will involve coding a software that operates on the blockchain.

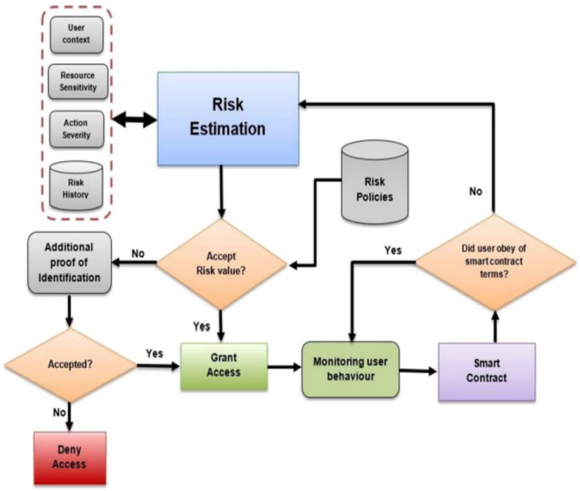

Fig. 15. Process flow of the proposed access control adaptive risk-based model-module 2(Odhiambo et al.,2024)

For each granted user in the proposed model, a smart contract shall be created. The monitoring module compares user behavior during access sessions with the contract's terms and conditions to identify abnormal actions. The user, through the access request, defines the data and actions, and if access is granted, the smart contract ensures adherence to these specifications. Monitoring includes validating accessed resources and actions against the contract's terms, issuing warnings and terminating sessions upon violations.

The proposed model aims to provide dynamic and adaptive secure access for smart contract execution. Incorporating real-time and contextual features into access decisions addresses unexpected circumstances, allowing for policy violations. Smart contracts play a crucial role in monitoring user activities, providing a significant solution for timely security violation detection, protecting system resources, and preventing sensitive information disclosure. While the risk estimation module is vital in risk-based models, the challenge lies in measuring security risks without a dataset describing the likelihood and impact of various incidents. Additionally, system flexibility is a key consideration when selecting a risk estimation technique.

In figure 15 a design algorithm process for the adaptive model smart contract blockchain-based systems is presented that monitors user activities using smart contract during access session. Blockchain will act as a software connector with a complex internal structure, with various configurations and different variants that are addressed herein the empirical frameworks. Finally, we then shall evaluate an algorithmic taxonomy of the adaptive risk based access model when granting access to a user for executing smart contracts. This taxonomy is intended to help with important architectural considerations about the identification of the user attributes continuously throughout the session.

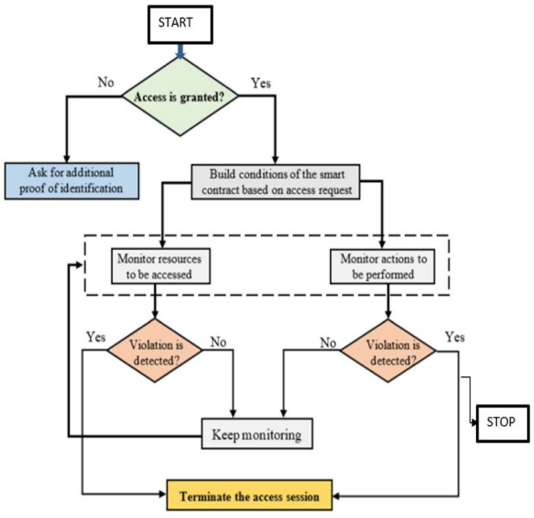

Fig. 16. Algorithmic monitoring user activities using smart contract during access session- module 3(Odhiambo et al.,2024)

1) The Combined Modules' Model

From the conceptual framework, our intention is to establish a link between the operationalization of the model and the studied dependent variables for the execution of smart contracts. This will be realized through the creation of an adaptive model utilizing the fuzzy logic fabric with expert judgment thematic mechanism.

To ensure secure access and the execution of smart contracts, it is crucial to establish a validation process through an algorithm such as proof of work or proof of stake. This consensus algorithm ensures agreement among all nodes on the block chain-network regarding the nature and state of the contract to be executed. Another essential component is the execution context of the smart contract, where the actual processing of contract terms, such as the transfer of digital assets and the update of records, occurs within lines of code. There are two different anonymity sets in a communication system: sender sets and recipient sets (Yaga et al.,2018). The adaptive risk-based access control model will also utilize Proof of Bandwidth (PoB) consensus mechanisms to grant access adaptively. In such cases, vast volumes of information need extraction and processing from databases for verification and validation to grant access into the system, involving agreements, documents, or other data sets.

In the transmission of desired transactions through nodes, a P2P network is involved. Through a recognized algorithm, the node network validates identity and user status. Subsequently, a new block is appended to the existing blockchain, containing a hash, verified proof of valid transactions consisting of a timestamp and the hash of the previous block. This prevents the block from being altered or a block being inserted between two existing blocks. Smart contracts executed based on certain conditions shall be encoded into the platform, applicable only to permissioned blockchains with a high degree of trust. After solving the proof of work (PoW) puzzle, the block will be broadcast to other nodes, detecting vulnerabilities and virtually preventing attacks from intruding. The main goal is to develop a less computational but adaptive risk-based access control model than PoW with better dynamic and robust access security guarantees. The publishing of the new node will depend on random waiting time from a secure hardware shell.

Constructing the proposed model involves the partitioning of data into training and validation sets. Training data are employed to derive the necessary rules for the block sets while validation data are utilized to assess the smart contract and implement-test required modifications. Represented as a flow diagram, the access decision features nodes, represented by rectangles, each describing the probability and impact of a risk. These rectangles are interconnected by arrows, with each arrow leading to another box indicating the percentage or threshold probability risk in accessing the block.

The model proves efficient in handling unexpected situations that may lead to policy violations due to imperfect policies. The model utilizes four inputs—user context, resource sensitivity, action severity, and risk history. These inputs inform the risk estimation module, responsible for assessing the overall risk value tied to each access request. Subsequently, the estimated risk value is compared with predefined risk policies to determine whether to grant or deny access. To enhance abnormality detection, we suggest incorporating smart contracts to track and monitor user activities during access sessions, mitigating potential malicious attacks and preventing sensitive information disclosure.

The proposed model incorporates real-time user context features, considering environmental attributes related to the user during access requests, such as location and time.

Resource/data sensitivity defines the importance of data susceptible to improper access, a subjective determination necessitating security experts' input for effective classification. Different types of data carry varying sensitivity levels, each assigned a sensitivity metric within the smart contract. When a user specifies actions on a resource, action severity gauges the impact on system resources, with security experts categorizing actions and assigning severity metrics with the aid of fuzzy logic inference system. This results in a risk metric associated with each action on a specific resource. User risk history captures past risk values for various user actions, aiding in distinguishing between good and malicious users. The risk estimation module, a crucial component, uses input risk factors to measure the risk value associated with each access request. The ultimate goal is to develop an efficient risk estimation method utilizing real-time information to precisely control access operations throughout the activity session of the user.

The estimated risk value undergoes comparison with risk policies to determine access decisions. Risk policies shall establish access boundaries and conditions for granting or denying access, defining a threshold value. If the risk value falls below this threshold, access is granted; otherwise, access is denied. The process flow of the proposed risk-based model, begins with a user sending an access request to the access control manager, specifying the resource or data to be accessed and the intended action. The access control manager then collects contextual information related to the user, including location and time, the sensitivity level of the resource, and the severity of the specified action, alongside the user's previous risk history records.

This process presents two scenarios. In the first scenario, where access is granted, smart contracts will play a pivotal role in tracking and monitoring user activities during the access session. The smart contract ensures adherence to contract terms and conditions, issuing warnings and terminating the session if any malicious activity is detected. The second scenario involves denying access, prompting the system to request additional proof of identification from the user to reduce false-positive rates. Correct credentials will result in access being granted, with subsequent monitoring, while incorrect credentials will lead to access denial. Classical access control models like cryptos' within block chain lack the ability to detect malicious actions and safeguard system resources after access is granted.

The risk estimation module dynamically adjusts user permissions based on behavior in access sessions, reducing privileges or terminating access sessions upon detecting abnormal actions. The monitoring module compares user behavior during access sessions with the contract's terms and conditions to identify abnormal actions. The user, through the access request, defines the data and actions, and if access is granted, the smart contract ensures adherence to these specifications. Monitoring includes validating accessed resources and actions against the contract's terms, issuing warnings and terminating sessions upon violations.

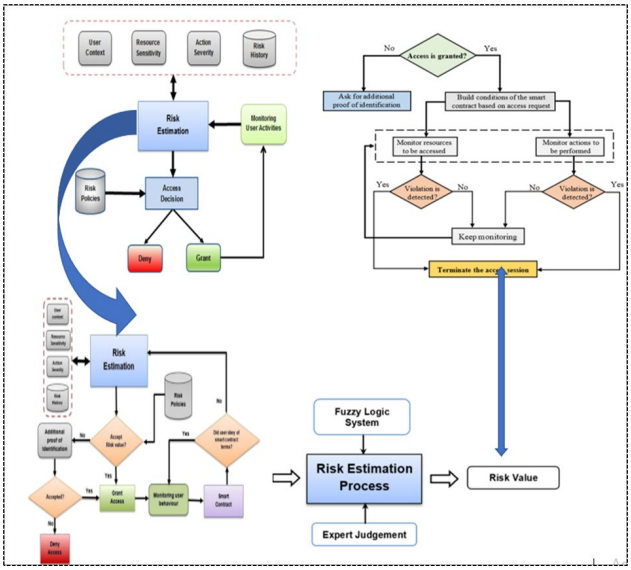

Fig. 17. Proposed adaptive risk-based access control architecture model-Full module (Odhiambo et al.,2024)

Figure 17 shows the proposed full module architectural design for the adaptive risk-based access control smart contract blockchain-based model that monitors user during access session. Blockchain, acts as a software connector with a complex internal modular structure, having various configurations and different variables that were earlier discussed that are intended for architectural considerations for the identification of the user attributes before and after granting access.

2) Pilot test

In this study, four distinct system modules will be devised and subsequently integrated to construct the adaptive risk-based access control model. The development of each module will involve both developers and users continuously evaluate the system to ensure its alignment with requirements and specifications. Additionally, developers will better understand the technical aspects and feasibility of the system. Programming the blockchain modules will entail writing code segments using the latest Waspmote Integrated Development Environment (IDE). The code will be uploaded and executed through the MATLAB toolkit, facilitated by a web interface. Waspmotes IDE by Libelium will be employed for this purpose.

The Fuzzy Freight and Logistics Business Smart Contract Knowledge System will be constructed using MATLAB to represent various facets of the smart contract execution phase. Regular updates and maintenance will be conducted to sustain the model's efficacy in addressing security risks. Exploratory data analysis and data visualization will be employed to uncover patterns and insights crucial for decision-making.

The modelling technique employed for our model will be machine learning algorithms or rule-based systems, chosen based on the complexity of the exogenous environment and the available data within the model's operation. The adaptive risk-based model will undergo continuous training using historical data with labeled risk levels. A portion of the data will be

allocated for training, with separate sets for testing and validation. Furthermore, adaptive learning mechanisms will be implemented to continuously update the model based on real-time data and evolving risk factors.

Table 8
Evaluation metrics and key performance indicators (Odhiambo et al.,2024)

Evaluation Metrics (EM)	Key-performance indicators (KPIs)	Specific parameters to evaluate the expected results	Evaluation design Methods
Security effectiveness	Number of security incidents detected, successful/unsuccessful access attempts, detection and response time to security threats.	Low level Moderate level High level	Quantitative methods: Surveys, experiments, MATLAB simulations, and analysis of system logs or performance metrics to collect quantitative data on security incidents, user perceptions, and system performance Qualitative methods: Focus groups-interview, usability testing, and observations to gather qualitative insights into user experiences, attitudes, and usability issues related to the adaptive model.
Adaptability	Ability of the model to dynamically adjust access control decisions based on changing risk levels, frequency of adaptation, accuracy of risk assessment.	Flexible Rigid	
Usability	Ease of configuration, comprehensibility of risk assessment factors, clarity of access control decisions, user-friendliness of the interface. User satisfaction perceptions of system administrators and end-users regarding the usability, intuitiveness, and effectiveness of the adaptive model.	Interactive Moderate Complex	

Monitoring and analyzing access patterns and security events will be integral to identifying changes in risk. Rigorous testing and validation will be conducted under various scenarios and conditions to pinpoint vulnerabilities or false positives/negatives. A feedback loop with stakeholders will be established to gather input and insights on the model's performance, leading to necessary adjustments.

Finally, incident response procedures will be developed for handling security incidents and breaches detected by the adaptive risk-based access control model. Periodic reviews and enhancements will be conducted to incorporate new data sources, technologies, and security best practices.

F. Evaluating the Adaptive Risk-Based Access Control Model

Research design plays a crucial role in evaluating an adaptive risk-based access control model by providing a systematic framework for collecting and analyzing data to assess its effectiveness, usability, and performance. Selection of evaluation metrics (EM) and key performance indicators (KPIs) that align with the objectives of the evaluation of the adaptive risk-based access control model shall be limited to as captured in the table 8

1) Reliability and Validity of the Instruments

The internal consistency of the items corresponding to each variable, as outlined in the conceptual framework, will be assessed by calculating the Cronbach's alpha coefficient using SPSS version 22. Cronbach's alpha is represented as a correlation coefficient that ranges from 0 to +1, measuring how well items in the instrument positively correlate with one another. The closer the estimated Cronbach's alpha coefficient is to 1, the higher the internal reliability of the instrument. DeVellis (2011) provides guidelines suggesting that an alpha coefficient above 0.7 is considered acceptable. If the items constituting the variable sets in the pilot study yield an average alpha of 0.80, this will be deemed satisfactory as it exceeds the 0.7 threshold.

The study will employ various types of data collection instruments, including checklists, observation and a system technical code test, along with the MATLAB toolkit, to introduce triangulation and enhance validity. To validate the tools, the researcher will seek the opinions of peers, supervisors, and experts in the fields of Information Systems and IT at the university. This approach aims to examine the content and assess the extent to which the instruments gather the intended information.

2) Evaluation through Model Analysis

The researcher will conduct data analysis using appropriate statistical or qualitative analysis techniques to derive meaningful insights and conclusions. Quantitative analysis may involve descriptive statistics, inferential statistics, regression analysis, and correlation analysis to examine relationships between variables and identify significant findings. Qualitative analysis will involve thematic analysis, content analysis, and interpretation of qualitative data to identify patterns, themes, and emerging insights related to user experiences and perceptions. Analysis of documents will entail a comprehensive scrutiny of relevant documents to extract pertinent information regarding the adaptive model and its components.

The model outcomes will undergo a comparative analysis with the outputs of existing models, and this comparison will be illustrated through graphical triangulation simulations. A multiple regression model will be employed to illustrate the degree of correlation between independent variables and the dependent variable according to the equation:

$$\mathbf{Y} = \beta_0 + \beta_1 \mathbf{X}_1 + \beta_2 \mathbf{X}_2 + \beta_3 \mathbf{X}_3 + \beta_4 \mathbf{X}_4 + \varepsilon$$

Where:

- Y represents smart contract consensus and execution context
- X_1 denotes scalability
- X_2 signifies interoperability
- X_3 refers to security legal and regulation
- X_4 represents identity management

In the model, β_0 stands for the constant term, and the coefficients β_i (ranging from 1 to 4) will gauge the sensitivity of the dependent variable (Y) to a unit change in the predictor variables (X_1 X_2 X_3 and X_4). The error term (ε) captures the unexplained variations in the model. The findings will be presented through charts, graphical simulation models, and tables designed for a user-friendly interface, ensuring easy interpretation.

This research methodology offers a structured approach to design, implement, monitor, and evaluate an adaptive risk-based access control model. This model aims to assist organizations in dynamically managing access permissions based on real-time risk assessments.

4. Conclusion

A. Significance of the Study

This study aims to address critical security gaps in smart contract execution on blockchain systems by proposing developing an adaptive risk-based access control model. Such a model enhances security by dynamically adjusting user permissions based on behavior during access sessions, thereby preventing unauthorized access and potential security breaches. The incorporation of a risk estimation module that adapts to user behavior reduces misuse and strengthens system integrity. The innovative use of fuzzy logic and expert judgment mechanisms introduces a proactive approach to access control, setting a new standard and making blockchain systems more robust and secure. The practical implications of this study are significant for industries relying on blockchain technology, including finance and supply chain management. By providing enhanced security, this model protects system resources from malicious actions during active access sessions that current models that rely on classical crypto access(grant or deny) fail to detect and mitigate.

The research outcomes can improve methodologies and theories for future studies, suggesting best practices for modeling, designing, and implementing adaptive risk models. Furthermore, this study not only fills a gap in existing research but also contributes valuable insights for both academic researchers and industry practitioners. The dual impact of the research enhances its relevance and importance, as findings can be directly applied to real-world blockchain systems, making them more secure, adaptive, and efficient. Overall, this research has the potential to revolutionize access control in blockchain systems, leading to more advanced and secure models in the future.

B. Limitations of the Study

Analyzing and mitigating risks in complex smart contract scenarios can be challenging and resource-intensive. The rapidly evolving blockchain technology, with new consensus mechanisms and protocol changes, may impact the model's effectiveness. Smart contracts, which can involve multiple conditions and dependencies, are vulnerable to security risks, requiring continuous assessment and addressing emerging threats. The scarcity of historical data due to the novelty of blockchain technology can hinder the development of robust risk models. Balancing transparency with privacy and ensuring regulatory compliance further complicate the development of adaptive risk models. The substantial computational resources required for smart contracts and adaptive risk models can impact system scalability and efficiency. Interoperability between different blockchain platforms is difficult due to varying standards and protocols. Adhering to diverse and evolving regulatory frameworks adds complexity to the development of risk access control models. The adoption of blockchain technology and smart contracts is still limited in certain industries, which can constrain research by reducing the diversity of use cases and applications.

Conducting research on adaptive risk access control models may face challenges such as user resistance to new access control mechanisms and reluctance to provide technical system information. Overcoming user education and adoption challenges is crucial for successful implementation. A comprehensive and multidisciplinary approach involving expertise in blockchain technology, cybersecurity, risk management, and regulatory compliance is essential. Researchers must stay updated with the latest developments and actively collaborate with industry stakeholders to address emerging challenges. The study acknowledges the potential difficulties in exhaustively covering all dimensions of a risk-based system and recommends further research to explore additional aspects.

The creation of innovative software applications on digital platforms represents a departure from conventional software development to dynamic platforms that address specific challenges within system applications (Leekwijck et al., 2019). Developers face the task of achieving an application-platform match, application-market match, providing value propositions that surpass the platform's core offerings, and delivering novelty. Digital virtual platforms introduce an environment marked by uncertainty, risk, and resource constraints, rendering

conventional approaches—such as plan-driven, ad-hoc, and controlled-flexible—of limited applicability (Harris et al., 2016).

Software development unfolds across four dimensions: technology, people, process, and product. A team of software development professionals, encompassing developers, testers, architects, designers, and project managers, must select appropriate technology (tools, programming language, hardware, software) for system development. However, this study is constrained by limitations in accessing all these resources simultaneously for the implementation-testing of the adaptive model.

C. Ethical Consideration for the Adaptive Risk-Based Access Control Model

Ethical considerations are paramount when designing, testing or implementing, and evaluating any technology, which is pertinent to our adaptive risk-based access control model. On the aspect of privacy and data protection, we ought to take into account and ensure that the model complies with privacy laws and regulations within the jurisdiction it operates. This necessitates the collection of only relevant data for risk assessment and access control decisions, and implement measures to protect sensitive information. We factor in provision of transparency to users regarding the data that will be collected and how these data will be used. Implement robust security measures to protect the adaptive model from unauthorized access, tampering, or exploitation. In our model we will endeavour to build trust with users by demonstrating the reliability, accuracy, and effectiveness of the adaptive risk-based access control model in making access control decisions autonomously and adaptively. This then affirms security and trustworthiness of our proposed model.

Guarding fairness and against biases in risk assessment algorithms that could lead to unfair treatment of certain individuals or groups by regular audits of the access control model to identify and mitigate biases that may arise from historical data or algorithmic decisions. In the evaluation phase we ought to ensure that users are adequately informed about the nature of the adaptive risk-based access control model, including how it dynamically adjusts access permissions based on risk assessments. Obtaining explicit consent from users before collecting and processing their data for risk assessment purposes will be considered.

Users' autonomy should be considered especially to allow control over own data and access permissions by which users can override or adjust access control decisions that are transacted by the adaptive risk-based access control model, in the instance they believe it is necessary or appropriate. Accountability and transparency will then ensure that decisions made by the model, including mechanisms for traceability and auditing provide transparency into the decision-making process and, on how risk assessments are conducted before decisions are made.

Minimization of potential harms or negative consequences resulting from the use of the model such as unjustified denial of access or exposure of sensitive information will account for continuous monitoring for unintended consequences or adverse effects of the adaptive model and if necessary, take prompt corrective actions. In the testing of the model, we will consider how best our model promotes equity, access and inclusivity by providing fair and equal access to resources and information for all users by avoiding exacerbating existing disparities or inequalities in access to technology and information.

Relevant permissions should be sought from NACOSTI, Maseno Ethical Review committee, and other Authorities that deal with data protection. Ultimately this then calls in for continuous evaluation and refinement of the adaptive risk-based access control model based on feedback from users, stakeholders, and ethical assessments of implications of the model's design, implementation or testing to ensure alignment with ethical principles and values. In lieu of addressing these ethical considerations throughout the lifecycle of the adaptive risk-based access control model, developers and practitioners then will promote responsible and ethical use of such technology while maximizing its benefits for individuals and firms.

References

[1] Abdelmaboud, A., Ahmed, A.I.A., Abaker, M., Eisa, T.A.E., Albasheer, H., Ghorashi, S.A., & Karim, F.K. (2022). Blockchain for IoT Applications: Taxonomy, Platforms, Recent Advances, Challenges and Future Research Directions. Electronics, 11, 630. https://doi.org/10.3390/electronics11040630

[2] Ahmed, E., Yaqoob, I., Hashem, I.A.T., Khan, I., Ahmed, A.I.A., Imran, M., & Vasilakos, A.V. (2017). The role of big data analytics in Internet of Things. Computer Networks, 129, 459–471.

[3] Ahubele, B., Eke, B., & Onuodu, F. (2021). On-Blockchain Validation Smart Contract Model on Ethereum Distributed Ledger System for Pharmaceutical Products Distribution. Journal of Computer Engineering, 23(2), 10–22.

[4] Aitzhan, N., & 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, 840–852.

[5] Arafat, S.M., Chowdhury, H.R., Qusar, M.S., & Hafez, M.S. (2016). Cross Cultural Adaptation & Psychometric Validation of Research Instruments: a Methodological Review. Journal of Behavioral Health, 5, 129–136.

[6] Arslan, S., Jurdak, R., Jelitto, J., & Krishnamachari, B. (2020). Advancements in Distributed Ledger Technology for Internet of Things. Elsevier: Amsterdam, The Netherlands.

[7] Atlam, H., Walters, R., Wills, G., & Daniel, J. (2021). Fuzzy Logic with Expert Judgment to Implement an Adaptive Risk-Based Access Control Model for IoT. Mobile Networks and Applications, 26. https://doi.org/10.1007/s11036-019-01214-w

[8] Azbeg, K., Ouchetto, O., Andaloussi, S., & Fetjah, L. (2021). A Taxonomic Review of the Use of IoT and Blockchain in Healthcare Applications. IRBM, in press.

[9] Aitken, S. (2016, April 5). Bitland's African Blockchain initiative putting land on the ledger. Forbes.

[10] Arafat, S. M., Chowdhury, H. R., Qusar, M. S., & Hafez, M. S. (2016). Cross Cultural Adaptation & Psychometric Validation of Research Instruments: a Methodological Review. Journal of Behavioral Health, 5, 129-136.

[11] Bangare, S., Gupta, M., Dalal, M., & Inamdar, A. (2016). Using Node.Js to Build High Speed and Scalable Backend Database Server. International Journal of Research in Advent Technology, 4(May), 19.

[12] Bore, S., Karumba, J., Mutahi, S., Darnell, S. S., Wayua, C., & Weldemariam, K. (2017). Towards Blockchain-enabled school information hub. In Proceedings of the 9th International Conference on Information and Communication Technology and Development.
[13] Bai, Y., Wang, D. (1982). Fundamentals of fuzzy logic control – fuzzy sets, fuzzy rules and defuzzifications. Advances in Fuzzy Logic Technologies in Industrial Applications, 17–36.
[14] Bancor. (2018). Bancor.
[15] Bankykoom. (2018). Blockchain powered solutions and services.
[16] Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A Software Architect's Perspective. Addison-Wesley Professional.
[17] Ben Dickson. (2018, January 30). Can blockchain democratize education? This startup seems to think so. The Next Web. Retrieved March 11, 2019.
[18] Binmore, K., & Vulkan, N. (2015). Applying game theory to automated negotiation. Economic Research Electronic Network, 1, 1–9.
[19] BitLand. (n.d.). Welcome to Bitland. Retrieved from
[20] Brodkin, J. (2008). Loss of customer data spurs closure of online storage service 'The Linkup'. Network World, August 2008.
[21] Buterin, V. (2018). A next-generation smart contract and decentralized application platform. White Paper, 3, 1–36.
[22] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 25(6), 599–616.
[23] Casino, F., Dasaklis, T. K., & Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. Telematics and Informatics, 36, 55–81.
[24] Castiglione, A., et al. (2016). Hierarchical and shared access control. IEEE Transactions on Information Forensics and Security, 11(4), 850–865.
[25] Cazzola, W., & Iannaccone, G. (2018). A privacy-preserving approach to user authentication in mobile cloud computing. Future Generation Computer Systems, 89, 142–153.
[26] Ceglowski, M. (2015). The Moral Economy of Tech. Retrieved from https://idlewords.com/talks/sase_panel.htm
[27] Chang, C. C., Fan, C. I., Lee, J. G., & Wu, K. M. (2018). A distributed approach to constructing an overlay network for P2P live streaming. IEEE Transactions on Multimedia, 10(8), 1675–1686.
[28] Chen, J., & He, H. (2016). IoT-based tracking system for medical supplies in smart hospital environment. Procedia Computer Science, 91, 1004–1011.
[29] Cheng, M., Zhang, H., Guo, L., & Sun, Z. (2016). Internet of things-based smart rehabilitation system. Journal of Sensors, 2016, 1–8.
[30] Chiang, M., Zhang, T., Wang, S., & Zhang, L. (2016). Fog and IoT: An overview of research opportunities. IEEE Internet of Things Journal, 3(6), 854–864.
[31] Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the Internet of Things. IEEE Access, 4, 2292–2303.
[32] Chui, M., Manyika, J., & Miremadi, M. (2016). Where machines could replace humans—and where they can't (yet). McKinsey Quarterly, July 2016.
[33] Coelho, R., Madureira, A. M., & Ribeiro, A. (2019). IoT-based recommender systems in tourism: A systematic literature review. Information Systems Frontiers, 21(2), 281–297.
[34] Chandrasekhar. (2018, April 6). The Emergence of Data Marketplaces. Hortonworks.
[35] Chandrasekhar. (2018, May 30). Blockchain-driven Data Marketplaces: A reference architecture. Hortonworks.
[36] Chandrasekhar. (2022, April 6). The Emergence of Data Marketplaces. Hortonworks.
[37] Carranza,E.J.M.,Porwal,A.& Hale,M.(2015). A hybrid neuro-fuzzy model for mineral potential mapping.Mathematical Geology.
[38] Dolgui, D., Ivanov, S., Potryasaev, B., Sokolov, M., Ivanova, M., & Werner, F. (2020). Blockchain-oriented dynamic modelling of smart contract design and execution in the supply chain. International Journal of Production Research, 58(7), 2184-2199.
[39] Diep, N., Hung, L. X., Zhung, Y., Lee, S., Lee, Y., & Lee, H. (2019). Enforcing access control using risk assessment. In Fourth European Conference on Universal Multiservice Networks, 419-424.
[40] Dai, W., Fan, K., & Ma, J. (2018). Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain. Journal of Medical Systems, 42(8), 1–9.
[41] Decker, C., & Wattenhofer, R. (2017). Information propagation in the Bitcoin network. IEEE P2P 2013 Proceedings, Trento, Italy.
[42] De Filippi, P., & Wright, A. (2018). Blockchain and the Law: The Rule of Code. Harvard University Press.
[43] Deng, R., Lu, R., Lai, C., Liang, X., & Shen, X. (2016). Optimal workload allocation in fog-cloud computing towards balanced delay and power consumption. IEEE Internet of Things Journal, 3(6), 1171–1181.
[44] Di Francesco Maesa, D., Capossele, A., Ghini, V., Marchetti, E., & Tombolini, R. (2021). A Blockchain and Fog Computing-Based Framework for Secure and Trusted Management of Smart Grids. IEEE Transactions on Industrial Informatics, 17(4), 2728–2737.
[45] Di Francia, G., Mariani, A., & Pagano, M. (2021). A Distributed Ledger Approach for Digital Manufacturing. IEEE Transactions on Industrial Informatics, 17(12), 8285–8294.
[46] Di Pietro, R., & Giordano, S. (2017). Data security in cloud storage systems: A survey. IEEE Communications Surveys & Tutorials, 19(2), 1035–1070.
[47] Dhillon, G., & Moores, T. (2001). Internet banking: An empirical investigation of adoption rates, consumer preferences, and attraction factors. International Journal of Bank Marketing, 19(7), 312–328.
[48] Dinh, T. T. A., Lee, C., Niyato, D., & Wang, P. (2017). A survey of mobile cloud computing: Architecture, applications, and approaches. Wireless Communications and Mobile Computing, 2017, 1–31.
[49] Dinh, T. T. A., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: Architecture, applications, and approaches. Wireless Communications and Mobile Computing, 13(18), 1587–1611.
[50] Dong, R., Zhang, C., & Zhao, Z. (2016). A cooperative coevolutionary algorithm with variable length chromosome representation for automated service composition. IEEE Transactions on Services Computing, 7(1), 2–14.
[51] Dorri, A., Kanhere, S. S., Jurdak, R., & Gauravaram, P. (2017). Blockchain for IoT security and privacy: The case study of a smart home. In 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) (pp. 618–623).
[52] Dorri, A., Kanhere, S. S., & Jurdak, R. (2018). Blockchain in internet of things: Challenges and solutions. In Proceedings of the 2018 International Conference on Blockchain (pp. 1–7).
[53] Dubey, A., Chaurasia, M., & Kumar, A. (2021). Blockchain-Based Secure Model for Healthcare System Using Homomorphic Encryption. In E. Tuncer, R. R. Mall, & S. M. Thampi (Eds.), Blockchain and Internet of Things for Secure, Scalable, and Efficient Frameworks (pp. 131–148). Springer.
[54] Durand, D., & Paquette, G. (2013). Applying knowledge management systems to learning systems. In C. S. Mumford (Ed.), Handbook of Organizational Learning and Knowledge Management (2nd ed., pp. 707–733). Wiley.
[55] Echeverria, J., & Riva, R. (2018). A comprehensive survey on fog computing: State-of-the-art and research challenges. Journal of Network and Computer Applications, 98, 27–42.
[56] Elsden, C., Manohar, A., Briggs, J., Harding, M., Speed, C., & Vines, J. (2017). Making sense of blockchain applications: A typology for HCI. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (p. 458). ACM.
[57] El-Telbany, M., & Hassanein, H. S. (2021). Fog computing and blockchain for smart healthcare. Computer Networks, 182, 107544.
[58] Eyal, I., & Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. In Proceedings of the 2014 Financial Cryptography and Data Security Conference (pp. 436–454). Springer.
[59] Faghih, R. T., Dahleh, M. A., & Chhatwal, J. (2017). A decentralized, scalable solution to the security, privacy, and interoperability of electronic health records. npj Digital Medicine, 1(1), 63.
[60] Farris, P. W. (2010). Competitive analysis: Concepts and techniques for analyzing industries and competitors. Simon and Schuster.
[61] Fernández-Caramés, T. M., & Fraga-Lamas, P. (2018). A review on the application of blockchain to the next generation of cybersecure industry 4.0 smart factories. IEEE Access, 6, 57478–57496.
[62] Ferrer, J. L., & Mazon, J. N. (2020). A blockchain-based solution for the secure storage of patient records. In A. Abraham, A. Hassanien, V. Snasel, & J. M. Munoz-Vargas (Eds.), Computational Intelligence in Information Systems (pp. 215–228). Springer.
[63] Franklin, L.R., & Festing L. (2012). Exploratory Experiments Philosophy of Science, 72 (December 2012), pp. 888-899.

[64] Filippoupolitis, A., & Gorbil, G. (2018). A blockchain-based secure logging system for wireless sensor networks. In Proceedings of the 5th IEEE International Conference on Cyber Security and Cloud Computing (pp. 65–70).
[65] Firdhous, M., & Rajapakse, D. (2018). An investigation into the potential use of blockchain technology for university certificates. In 2018 3rd International Conference on Computing, Communication and Security (ICCCS) (pp. 1–6).
[66] Firdhous, M., & Rajapakse, D. (2019). A blockchain-based approach for secure handling of university student records. In 2019 Moratuwa Engineering Research Conference (MERCon) (pp. 175–180).
[67] Firdhous, M., & Rajapakse, D. (2020). A blockchain-based solution for managing student records securely. In 2020 Moratuwa Engineering Research Conference (MERCon) (pp. 16–21).
[68] Firdhous, M., & Rajapakse, D. (2021). A blockchain-based framework for secure storage and sharing of academic credentials. In 2021 Moratuwa Engineering Research Conference (MERCon) (pp. 1–6).
[69] Firdhous, M., Rajapakse, D., & Kulatunga, C. (2019). A blockchain-based approach for secure sharing of electronic health records in cloud computing. In 2019 Moratuwa Engineering Research Conference (MERCon) (pp. 369–374).
[70] Firdhous, M., Rajapakse, D., & Kulatunga, C. (2021). A blockchain-based framework for secure sharing of electronic health records in cloud computing. International Journal of Advanced Computer Science and Applications, 12(2), 1–13.
[71] Fraim, M., & Jones, S. (2018). Exploring blockchain's potential for vertical integration in the supply chain. Journal of Corporate Accounting & Finance, 29(6), 123–130.
[72] Fu, H., Xu, X., & Mei, Y. (2020). Blockchain-based secure and privacy-preserving data sharing scheme for IoT. IEEE Internet of Things Journal, 7(6), 4961–4971.
[73] Fu, Y., Wu, Y., Zhu, H., & Zhang, H. (2018). A blockchain-based medical prescription authentication scheme in collaborative healthcare environments. Future Generation Computer Systems, 86, 405–413.
[74] Gai, K., & Qiu, M. (2018). Blockchain in healthcare: A patient-centered model. In 2018 IEEE International Conference on Smart Cloud (SmartCloud) (pp. 243–248).
[75] Ganz, F., Barnaghi, P., Carrez, F., & Gyrard, A. (2018). A benchmarking framework for the performance evaluation of stream processing platforms for IoT applications. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 4165–4174).
[76] Gao, L., & Lu, R. (2019). Blockchain-based secure energy trading mechanism in industrial internet of things. IEEE Transactions on Industrial Informatics, 16(6), 3972–3980.
[77] Gao, L., Lu, R., & Liang, X. (2020). A blockchain-based privacy-preserving incentive mechanism for mobile crowdsensing systems. IEEE Transactions on Dependable and Secure Computing, 17(5), 1173–1185.
[78] Gaur, A., Sengupta, S., Sharma, M., Buyya, R., & Kapoor, S. (2019). Blockchain-enabled smart contracts: Architecture, challenges, and future trends. Future Generation Computer Systems, 95, 471–491.
[79] Gazi, P., Polychronakis, M., & Keromytis, A. D. (2018). Scriptless attacks: Stealing the pie without touching the sill. In 27th USENIX Security Symposium (USENIX Security 18) (pp. 1757–1774).
[80] Gartner. (2018). Gartner Top 10 Strategic Technology Trends for 2018.
[81] Ge, M., Xu, D., Ren, J., & Wang, Q. (2019). A blockchain-based framework for trustworthy data sharing in a cloud environment. Future Generation Computer Systems, 92, 232–240.
[82] Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., & Capkun, S. (2016). On the security and performance of proof of work blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 3–16).
[83] Ghose, S., Adil, M. A., & Han, Q. (2019). A novel consensus protocol for blockchain-based IoT devices. IEEE Internet of Things Journal, 6(2), 2920–2929.
[84] Ghosh, A., Guleria, K., Mohania, M., & Mohan, C. (2019). Blockchain-based data integrity for IoT data streams. In 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 157–165).
[85] Ghosh, A., Guleria, K., Mohania, M., & Mohan, C. (2019). DIBS: A decentralized and immutable blockchain based data integrity system for IoT. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 4432–4441).
[86] Gong, J., Zhang, K., Ma, J., & Xu, L. (2018). Blockchain-based data sharing: A survey. Journal of Internet Technology, 19(5), 1457–1466.
[87] Gong, Y., & Liu, H. (2018). Research on information security based on blockchain in the era of big data. Journal of Physics: Conference Series, 1069(1), 012048.
[88] Gu, W., Zhu, Z., Sun, Z., Wang, H., & Yu, F. R. (2020). A survey on consensus mechanisms and mining strategies for blockchain networks. IEEE Access, 8, 191487–191516.
[89] Guan, Y., Wu, X., Wang, Y., & Zhang, Z. (2019). Blockchain-based identity authentication mechanism for IoT. In 2019 15th International Conference on Computational Intelligence and Security (CIS) (pp. 25–29).
[90] Guo, Q., Zhang, L., Chen, H., & Zomaya, A. Y. (2019). Blockchain-based data preservation system for cloud storage. In 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS) (pp. 270–281).
[91] Guo, S., Guo, X., Li, F., & Li, Z. (2018). A survey of blockchain consensus algorithms. Journal of Computer Research and Development, 55(9), 2022–2039.
[92] Guo, T., Deng, R. H., Li, Z., & Yu, Y. (2018). Lightweight RFID mutual authentication protocol based on blockchain. IEEE Internet of Things Journal, 5(5), 3921–3928.
[93] Gupta, M., Jain, R., & Jain, S. (2018). A blockchain-based approach to secure IoT data. In 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 615–619).
[94] Haddad, P., & Dagher, G. G. (2020). Survey of blockchain technologies for Internet of Things. IEEE Access, 8, 26442–26463.
[95] Harris, M. L., Collins, R. W., & Hevner, A. R. (2016). Control of Flexible Software Development Under Uncertainty. Information Systems Research, 20(3), 400-419.
[96] Huang, X., Li, J., Chen, X., & Xiang, Y. (2018). Securely outsourcing attribute-based encryption with checkability. IEEE Transactions on Parallel and Distributed Systems, 25(8), 2201-2210.
[97] Han, C., Zhang, J., He, J., & Zhao, X. (2020). Blockchain-based data sharing and access control mechanism for industrial Internet of Things. IEEE Access, 8, 62401–62414.
[98] Han, L., Gao, Z., Wang, Y., Sun, W., & Yang, Y. (2019). Blockchain-based secure data storage and sharing scheme for industrial Internet of Things. In 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS) (pp. 448–453).
[99] Han, X., Li, C., & Zhou, M. (2019). A blockchain-based access control system for the Internet of Things. Journal of Network and Computer Applications, 134, 1–12.
[100] Han, X., Zhang, Y., Li, C., & Yang, Y. (2019). A blockchain-based approach to secure and trustworthy data sharing in fog-supported IoT. Journal of Network and Computer Applications, 138, 41–48.
[101] He, J., Ye, Y., Zhu, J., & Cao, Z. (2019). Blockchain-based secure data sharing of IoT in smart grid. IEEE Access, 7, 13450–13458.
[102] Han, X., Yang, Y., & Li, C. (2020). A decentralized access control scheme for blockchain-based Internet of Things. Journal of Parallel and Distributed Computing, 140, 119–129.
[103] Hatzivasilis, G., Nicopolitidis, P., Obaidat, M. S., Karyotis, V., & Logothetis, M. (2019). Secure IoT environments using blockchain: Opportunities and challenges. Computer Networks, 159, 106–124.
[104] He, D., & Chen, X. (2019). A novel blockchain-based data integrity protection framework for IoT data in smart cities. IEEE Transactions on Industrial Informatics, 15(3), 1675–1682.
[105] Hossain, M. S., Muhammad, G., & Ma, J. (2019). Blockchain-based secure data sharing of IoT: A systematic literature review, taxonomy and future directions. Journal of Network and Computer Applications, 125, 134–153.
[106] Hu, L., Zhang, H., Jiang, P., & Qian, Y. (2020). A secure access control scheme for blockchain-based IoT systems. Future Generation Computer Systems, 105, 450–461.
[107] Hu, S., & Xu, X. (2019). Blockchain-based data sharing security scheme for industrial Internet of Things. IET Networks, 8(3), 133–140.
[108] Huang, Y., Chen, X., & Liu, J. K. (2020). A blockchain-based secure data sharing scheme for IoT systems. Future Generation Computer Systems, 110, 721–729.
[109] Hwang, J. H., & Choi, S. G. (2019). A blockchain-based secure data sharing scheme using a smart contract for IoT. Electronics, 8(10), 1137.

[110] Hwang, J. H., & Kim, H. (2020). A secure data sharing scheme based on blockchain for IoT. IEEE Access, 8, 40940–40949.
[111] IBM Corporation. (2018). IBM and Maersk form global joint venture applying Blockchain to shipping logistics.
[112] Islam, S. R., & Chang, V. (2018). Smart contract enabled access control for the internet of things. Future Generation Computer Systems, 86, 1046–1059.
[113] Islam, S. R., Kwak, D., Kabir, M. H., Hossain, M. A., & Kwak, K. S. (2019). The Internet of Things for health care: A comprehensive survey. IEEE Access, 7, 64729–64749.
[114] Islam, S. R., & Kwak, K. S. (2018). Blockchain-based secure firmware update for embedded devices in an Internet of Things environment. Journal of Network and Computer Applications, 116, 42–52.
[115] Jia, K., Tang, S., & Xu, J. (2020). A blockchain-based secure data sharing scheme for Industrial Internet of Things. Journal of Network and Computer Applications, 170, 102722.
[116] Kortesniemi,Y., Mikko, S.(2014). Survey of certificate usage in distributed access control doi:10.1016/j.cose.2014.03.013
[117] Leekwijck, W. V., & Kerre, E. E. (2019). Defuzzification: criteria and classification. Fuzzy Sets and Systems, 108(2), 159-178.
[118] MathWorks. (2021). MATLAB - MathWorks. Retrieved from
[119] Odhiambo.A.,Oteyo E., & Oonge, S.(2024). Unpublished Adaptive risk based access control model design for smart contract execution on blockchain systems, Maseno University.
[120] Rajbhandari, L., & Snekkenes, E. A. (2016). Using game theory to analyze risk to privacy: an initial insight. In Privacy and Identity Management for Life (pp. 41-51). Springer Berlin Heidelberg.
[121] Ruddick, W. (2016). Eco-Pesa: An Evaluation of a Complementary Currency Programme in Kenya's Informal Settlements. International Journal of Community Currency Research, 15(A), 1-12.
[122] Santos, D. R., Westphall, C. M., & Westphall, C. B. (2019). A dynamic risk-based access control architecture for cloud computing. In IEEE/IFIP Network Operations and Management Symposium (NOMS), 1-9.
[123] Tiwana, A. (2013). Platform Ecosystems. Morgan Kaufmann.
[124] Tilson, D., Lyytinen, K., & Sorensen, C. (2017). Digital Infrastructures: The Missing IS Research Agenda. Information Systems Research, 21(4), 748-759.
[125] Watanabe, H., Fujimura, S., Nakadaira, A., Miyazaki, Y., Akutsu, A., & Kishigami, J. (2016). Blockchain contract: securing a blockchain applied to smart contracts. 2016 IEEE International Conference on Consumer Electronics, 467-468.
[126] Windley. (2018, January 10). How Blockchain makes self-sovereign identities possible. Computer World.
[127] Yaga, P., Mell, N., Roby, N., & Scarfone, K. (2018). Blockchain Technology Overview. National Institute of Standards and Technology Internal Report 8202, 1-66.
[128] Yin, J., Tang, C., Zhang, X., & McIntosh, M. (2016). On estimating the security risks of composite software services. In First Program Analysis for Security and Safety Workshop Discussion.
[129] Zhang, G., & Parashar, M. (2017). Dynamic context-aware access control for grid applications, 101-108. 10.1109/GRID.2003.1261704.
[130] Zhang, H.-R., Min, F., He, X., & Xu, Y.-Y. (2015). A Hybrid Recommender System Based on User-Recommender Interaction. Mathematical Problems in Engineering, 2015, Article ID: 145636.
[131] Zadeh, L. A. (2015). The concept of a linguistic variable and its applications to approximate reasoning. Information Sciences, 8(4), 199-249.