BLOCKCHAIN™ IN HEALTHCARE TODAY

ORIGINAL RESEARCH

S^{3}EF-HBCAs: Secure and Sustainable Software Engineering Framework for Healthcare Blockchain Applications

Muthu Ramachandran, PhD*

AI Tech, Leeds, England and Forti5 Tech, London, United Kingdom

*Corresponding Author: Muthu Ramachandran, Email: muthuram@ieee.org

Keywords: blockchain architecture, ethics, privacy, engineering for blockchain, framework for blockchain, security, smart contract, software framework, sustainability

Abstract

Blockchain applications in healthcare have grown rapidly. They include record-keeping, clinical trials, medical supply chains, patient monitoring, etc., where blockchain characteristics are needed to improve safety, privacy, and security. Blockchain technology is one of the most significant disruptive technologies today. However, Porru et al.[1] reported that it lacks processes, tools, and techniques. Therefore, this paper provides a systematic framework for a secure and sustainable software engineering framework for healthcare blockchain applications (S^{3}EF-HBCA). S^{3}EF-HBCA is a significant contribution that includes requirements engineering for healthcare, business process modeling for healthcare, domain modeling for healthcare, a reference architecture for healthcare, and validation by a case study on electronic healthcare record management system (EHR), and simulation with business process modeling notation (BPMN) tools. The simulation shows it has taken 10.45 min to process 100 instances of real-time data and service requests. The overall result shows encouragement regarding process, tools, standards, and testing.

Submitted: October 18, 2023; Accepted: December 3, 2023; Published: December 22, 2023

A summary of the main highlights of this article is listed here.

1. Contribution to designing blockchain applications in healthcare with built-in security, reusability, and sustainability.
2. Innovative approach to building a Secure and Sustainable Software Engineering Framework for Healthcare blockchain Applications (S3EF-HBCA) framework.
3. Requirements for engineering framework using business process modeling notation (BPMN) modeling and simulation for verification and validation before design and implementation.
4. Non-functional requirements classification for blockchain applications in healthcare.
5. Domain analysis method and classification for identifying and developing reusable smart contracts in healthcare applications.
6. Blocks-Point Effort Estimation Technique, including TCF and Environment Complexity Factors (ECF) for cost and complexity analysis.
7. Design strategies for building reusable blockchains as service components.
8. Innovate reference architecture (REF-ArcBC) standardizes blockchain development in healthcare applications.
9. Business process and reference architecture efficiency estimation metrics.

This detailed report delves into the motivation behind the rise of blockchain, strategic approaches to its implementation, and its current growth trends.

Blockchain has impacted medicine as well as other disciplines in many ways. For example, blockchain has been adopted rapidly in most applications today as it offers trust and security. The Computing Technology Industry Association (CompTIA)[2] defines blockchain as 'a mathematical structure for storing digital transactions or data in an immutable, distributed, decentralized digital ledger consisting of blocks that are linked via cryptographic signature that is nearly impossible to fake, hack or disrupt.' Banafa[3] reported that blockchain applications will generate $3.1 trillion in new business value by 2030. In addition, Banafa[3] also argues that 'The basis for a dynamic distributed ledger can be applied to save time when recording transactions between parties, remove costs associated with intermediaries, and reduce risks of fraud and tampering.'

Blockchain technology is a disruptive force, reshaping industries and revolutionizing transactions and data management. Its core principles of decentralization, transparency, security, and immutability have motivated organizations across various sectors to explore and adopt blockchain solutions.

Blockchain applications in healthcare have expanded rapidly, and they include record keeping, clinical trials, medical supply chains, patient monitoring, etc., where blockchain characteristics are needed to blockchain has been adopted rapidly in most applications today as it offers trust and security. CompTIA[2] defines blockchain as 'a mathematical structure for storing digital transactions or data in an immutable, distributed, decentralized digital ledger consisting of blocks that are linked via cryptographic signature that is nearly impossible to fake, hack or disrupt.' Banafa[3] reported that blockchain applications will generate $3.1 trillion in new business value by 2030. In addition, Banafa[3] also argues that 'the basis for a dynamic distributed ledger can be applied to save time when recording transactions between parties, remove costs associated with intermediaries, and reduce risks of fraud and tampering.'

Overall, blockchain technology in healthcare continues to grow, driven by the industry's need for data security, transparency, and efficiency. These metrics highlight the increasing adoption of blockchain solutions across various healthcare applications, promising a transformative impact on the sector in the coming years. As of today, blockchain technology continues to experience robust growth and development:

- Increased Adoption: Businesses across industries, including finance, supply chain, healthcare, and real estate, are actively adopting blockchain solutions to enhance efficiency and transparency.
- Evolving Ecosystem: The blockchain ecosystem is continuously evolving with the emergence of new blockchain platforms, projects, and protocols, each catering to specific use cases and requirements.
- Cryptocurrency Market: Cryptocurrencies, such as Bitcoin and Ethereum, continue to gain mainstream acceptance as alternative investments and digital assets.
- DeFi and NFTs: The decentralized finance (DeFi) and non-fungible token (NFT) markets have exploded, showcasing the versatility and potential of blockchain technology.
- Regulatory Developments: Governments and regulatory bodies are actively working to create a regulatory framework for blockchain and cryptocurrencies, aiming to balance innovation with consumer protection.
- Institutional Involvement: Institutional investors and financial institutions are increasingly participating in the cryptocurrency and blockchain space, indicating growing confidence in the technology.
- This research is based on an experimental case study on HER and doesn't involve any datasets.

In conclusion, the growth of blockchain technology is driven by its ability to provide trust, transparency, and security in various applications. Organizations are strategically adopting blockchain to innovate, optimize processes, and remain competitive in an ever-changing digital landscape. As the ecosystem matures and regulatory clarity improves, blockchain's role in shaping the future of industries is poised to expand further.

Destefanis et al.[4] reported a need for smart contract programming based on a disciplined approach. They reported several vulnerabilities causing the freezing of more than 500K users with a loss of $150 million. In addition, studies on software engineering (SE) for blockchain Dapps sought a systematic approach [Beller and Hejderup[5]; Destefanis et al. [4]; Chung L, do Prado Leite JCS[6]]. These articles focus on exploring vulnerabilities in smart contracts and emphasize the need for proper SE practices in the context of blockchain technology.

Smart contracts are self-executing contracts with predefined rules and conditions encoded on a blockchain. They are an integral part of many blockchain platforms, such as Ethereum. Smart contracts are powerful tools, but they are not immune to vulnerabilities. The article discusses various security issues that can arise in smart contracts, such as coding errors, design flaws, and implementation issues. These vulnerabilities can potentially lead to financial losses, exploitation, or disruptions in blockchain applications.

The author argues that traditional SE practices should be applied to developing and testing smart contracts to enhance their security and reliability. They may suggest specific approaches or methodologies for ensuring the correctness and robustness of smart contracts.

Overall, the articles highlight the importance of addressing vulnerabilities in smart contracts and advocate for adopting best practices from SE to improve the quality and security of blockchain-based applications. What do you gain from this article (learning outcome) and highlighting some of the key research questions?

The organization is as follows: Introduction provides key questions and learning outcomes; a critical evaluation of the literature survey on blockchain in healthcare today, key challenges and characteristics of blockchain, the S^{3}EF-HBCAs—Secure and Sustainable SE Framework, requirements engineering model for blockchain known as Blockchain Security Quality Requirements Engineering (BC-SQUARE) which is the key to achieving quality development of blockchain by analyzing and evaluating healthcare customer and stakeholders requirements, design method and a reference architecture known as REF-ArchBC which is the key to achieve standardization of blockchain apps, and evaluation techniques and best practices of reference architecture with a real-world case study.

Key Research Questions

- What are the key challenges in blockchain for healthcare applications?
- How do we systematically apply blockchain to be secure, safe, and sustainable technology in healthcare applications?
- What are the design principles for a driven reference architecture for a secure, sustainable, and software engineering approach to healthcare blockchain applications (S^{3}EF-HBCAs)?
- What are the services comprising reference architecture for S^{3}EF-HBCAs in healthcare?
- How to classify technologies and services for blockchain applications?
- How do we classify the application domain for building sustainable and reusable blockchain in healthcare?
- What are the key challenges of sustainability in S^{3}HBCAs mean?
- What are the key challenges of security in HBCAs?

Learning Outcomes

- Understand the current trends in healthcare blockchain applications (HBCAs)
- Identify some of the key characteristics of blockchain relevant to HBCAs
- To understand the security, sustainability, and software engineering approach in the HBCAs Framework (S3EF-HBCAs)
- Understand Requirements Engineering for S3EF-HBCAs
- Acquire knowledge of Domain Classification for Blockchain Applications
- Understand Blocks-Points Effort and Complexity Estimation
- Understand Reference Architecture for Blockchain in S3EF-HBCAs
- Understand BPMN modeling and simulation using a Case Study and Evaluation of REF-ArcBC with BPMN: Electronic Healthcare Record (EHR)
- Understand the Evaluation Techniques of BPMN simulation for Reference Architecture for S3EF-HBCAs.

Background and Critical Evaluation

Blockchain technology is a distributed ledger technology that gained prominence with the advent of cryptocurrencies like Bitcoin. It has since found applications in various industries, including healthcare. The core concept behind blockchain is the creation of a decentralized and immutable ledger that records transactions securely and transparently. In healthcare, this technology promises to enhance data security, interoperability, and transparency. Figure 1 shows the benefits of blockchain applications in healthcare, including key blockchain characteristics such as decentralization, improved data security, and privacy, as blockchain provides trust, health data ownership, availability, robustness, integrity, transparency and trust, and data verifiability.

Agbo, Mahmoud, and Eklund[7] discuss the potential applications of blockchain in healthcare and its benefits in terms of security and privacy and also provide a list of benefits of blockchain to healthcare applications: Decentralization, improved data security and privacy, health data ownership, availability/robustness, transparency and trust, and data verifiability. Mayer et al.[8] explore the use of blockchain technology for managing electronic health records and discuss the challenges associated with privacy. Christidis and Devetsikiotis[9] provide insights into the concept of smart contracts and their application in various domains, including healthcare. Khezr et al.[10] discuss different blockchain applications in healthcare, such as supply chain management and health records. Hasselgren et al.[11] explore blockchain technology's challenges and use cases in healthcare.

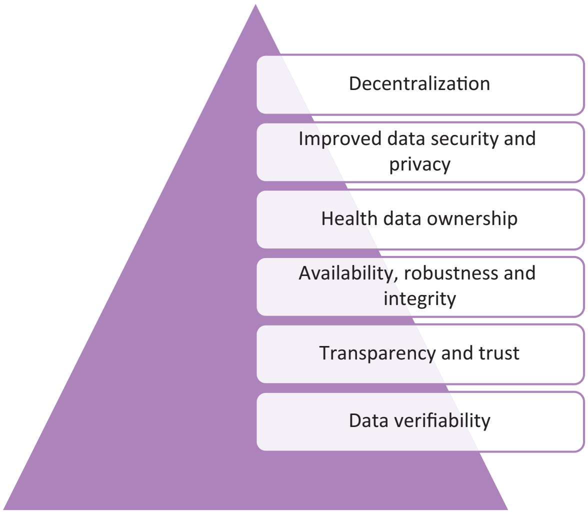


Fig. 1. Benefits of blockchain in healthcare applications.

Besides several benefits, blockchain technology also poses some key challenges, including ethical and legal issues. Ethical considerations on blockchain technology are crucial to ensure patient privacy, data security, and overall fairness. For example, on data privacy and consent, we must consider the following aspects:

- Obtain explicit and informed patient consent for data sharing and storage on the blockchain.
- Ensure compliance with data protection regulations like GDPR, HIPAA, or other local laws.

Tang et al.[12] provide a systematic discussion on the ethics of blockchain applications and map the main social challenges raised by its technology and applications. De Filippi and Wright[13] provide a detailed discussion on the legal aspects of blockchain concerning code as a rule versus code as law (blockchain), as smart contracts promise to support decentralized autonomous organizations (DAO) by automating the law through code. This creates a major fear amongst lawyers, lawmakers, and governments alike. In addition, there are major strengths and research challenges on security and privacy, interoperability, building reusable smart contracts for sustainability and productivity, and scalability.

Sustainability is an important factor for blockchain technology. This paper defines *sustainability in terms of several factors: reusability of the blockchain, such as smart contract,* as it has the potential to reduce energy consumption across the blockchain network, reproducibility of the results, energy consumption, social impact on the lifestyle improvement, and economic impact on the society. To this end, Giungato et al.[14] have reported that transitioning monetary systems to cryptocurrency will result in unacceptable energy consumption. They also reported that 25 new bitcoins are being generated every 10 min globally. This amounts to difficulty and the complex process of mining bitcoins involves a larger number of complex networks, PCs, virtual memory, etc. The following section identifies some of the key characteristics of blockchain and its significant contribution to dApps development, process, and identifying requirements.

Characteristics of Blockchain

Identifying the characteristics of blockchain is essential for understanding its potential business opportunities, applications, advantages, and limitations. Blockchain technology has gained significant attention due to its unique attributes, which set it apart from traditional databases and ledgers. Here are some key rationales for identifying the characteristics of blockchain:

1. Innovation and Disruption: Blockchain is a relatively new technology that can potentially disrupt various industries, including finance, supply chain, healthcare, and more. Understanding its characteristics helps businesses and individuals assess how it can be used to innovate and improve existing processes.
2. Security: Blockchain is often touted for its security features, including cryptographic encryption and immutability. Recognizing these characteristics is crucial for evaluating its suitability in applications where data security and integrity are paramount, such as financial transactions or medical records.
3. Transparency and Trust: Blockchain's transparent and decentralized nature allows multiple parties to trust a single source of truth without relying on intermediaries. This transparency is important in industries like supply chain management, where stakeholders need to trace the origin and journey of products.
4. Decentralization: Blockchain operates on a decentralized network of nodes, eliminating the need for a central authority. This characteristic can reduce the risk of a single point of failure, censorship, or manipulation, making it relevant in scenarios where trust in centralized institutions is eroded.
5. Immutability: Once data is added to a blockchain, it is challenging to alter or delete. This immutability can be advantageous in scenarios that require an unchangeable record, such as land registries or legal contracts.
6. Smart Contracts: Blockchain platforms like Ethereum enable the creation and execution of smart contracts, self-executing agreements with predefined rules. Identifying this characteristic is essential for exploring automation possibilities in various industries.
7. Consensus Mechanisms: Blockchain networks rely on consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS) to validate and add new transactions to the chain. Understanding these mechanisms is crucial for assessing the network's security, scalability, and energy efficiency.
8. Cryptocurrency: Many blockchains have their native cryptocurrencies (e.g. Bitcoin, Ethereum's Ether). Recognizing this characteristic is important for comprehending the role of digital assets in the blockchain ecosystem, including their use for payments, incentives, or governance.
9. Scalability and Performance: Different blockchain platforms have varying scalability and performance characteristics. Identifying these factors helps organizations select the right blockchain for their specific use cases and understand its limitations.
10. Privacy: While blockchains are often associated with transparency, some offer privacy features, such as confidential transactions or zero-knowledge proofs. Recognizing these privacy characteristics is vital for

industries like healthcare or finance, which require data confidentiality.

11. Energy Efficiency: Blockchain's energy consumption has been a topic of debate. Identifying this characteristic is important for making informed decisions about its environmental impact and sustainability.
12. Regulatory and Legal Implications: Understanding blockchain characteristics helps regulators and legal authorities address legal and compliance issues related to its use, such as data protection, taxation, and liability.

In summary, identifying the characteristics of blockchain is essential for making informed decisions about its adoption, assessing its suitability for specific use cases, and navigating the evolving landscape of blockchain technology and its applications. It allows businesses, policymakers, and individuals to harness the benefits of blockchain while addressing its challenges effectively. Figure 2 shows the core concepts of blockchain.

The core concepts of blockchain include a distributed system of record (blocks), security, verifiability and provenance, embedded business terms (smart contracts), and consensus and agreement. In addition, Viriyasitavat and Hoonsopon[15] state that some of the key characteristics of integrity, transparency, and resiliency are attractive to modern business process management (BPM), Service Workflow, Internet of Things (IoT), Industrial Internet of Things (IIoT), Healthcare, Cloud Computing, Big Data, Cyber-Physical Systems (CPS), etc.

Hakak et al.[16] have identified four key characteristics of blockchain that are relevant for building smart cities: consensus helps to prevent fraudulent transactions, transparency helps to establish easy validation of information, robustness helps to establish no single point of failure, and incorruptible helps to establish secured information and to build trust which is paramount and promise of blockchain. De Filippi and Wright[13] discuss some of the characteristics of blockchain that exhibit a set of core characteristics such as peer-to-peer networking, public-private key cryptography, and consensus mechanisms.

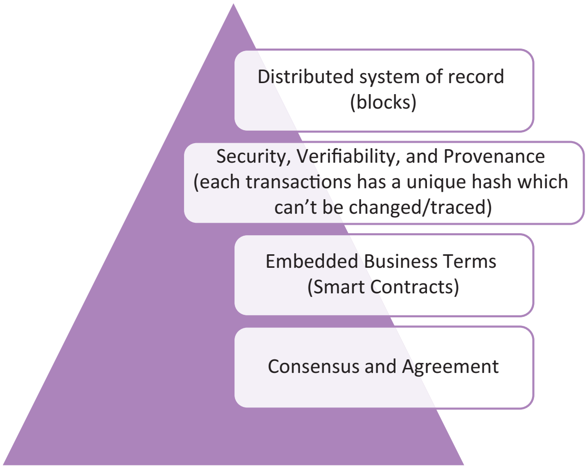

Distributed system of record (blocks)
Security, Verifiability, and Provenance (each transactions has a unique hash which can't be changed/traced)
Embedded Business Terms (Smart Contracts)
Consensus and Agreement

Fig. 2. Core concepts of blockchain.

Figure 3 provides a set of organized characteristics of blockchain and its benefits for healthcare applications. These include data mining, immutability, blockchain as software connected to emerging technologies such as data science and AI, blockchain data structures, and its limitations on scalability. For example, mainstream public blockchains can only handle an average of 3 to 20 transactions per second, whereas mainstream payment services, like VISA, can handle an average of 1,700 transactions per second. Therefore, it is important to consider closely the requirements for dApp development on the security and privacy of blockchain.

In conclusion, in this section, the characteristics of blockchain play a major role in fine-tuning requirements for the development of dApp healthcare applications, and it is important to adopt a systematic approach based on the SE framework presented in the following sections.

S^{3}ef-Bc: Secure and Sustainable Software Engineering Framework for Blockchain Applications

In the ever-evolving landscape of evolving modern and integrated technologies such as AI, machine learning, and data science, blockchain has emerged as a transformative force with the potential to revolutionize industries ranging from finance and supply chain management to healthcare and beyond. This decentralized and tamper-resistant ledger technology has garnered significant attention for its promise to enhance transparency, security, and efficiency in various domains. However, deploying blockchain applications also brings forth unique challenges, particularly regarding security and sustainability. As we venture deeper into the digital age, the need for a robust framework that ensures the security of blockchain systems and promotes their sustainability becomes increasingly paramount. In response to this imperative, the concept of a Secure and Sustainable Software Engineering Framework for Blockchain Applications has emerged as a critical area of focus. This framework seeks to establish best practices, methodologies, and tools that not only fortify the security of blockchain-based solutions but also address the environmental and resource-intensive concerns associated with blockchain technology. In this exploration, we delve into this framework's core principles and components, shedding light on its significance in shaping the future of blockchain applications. A software framework provides a structure to follow from requirements to deployment. Two fields of specialization have emerged: *Software engineering for Blockchain (SE for BC) and Blockchain for Software Engineering (BC for SE)*. Beller and Hejderup[5]

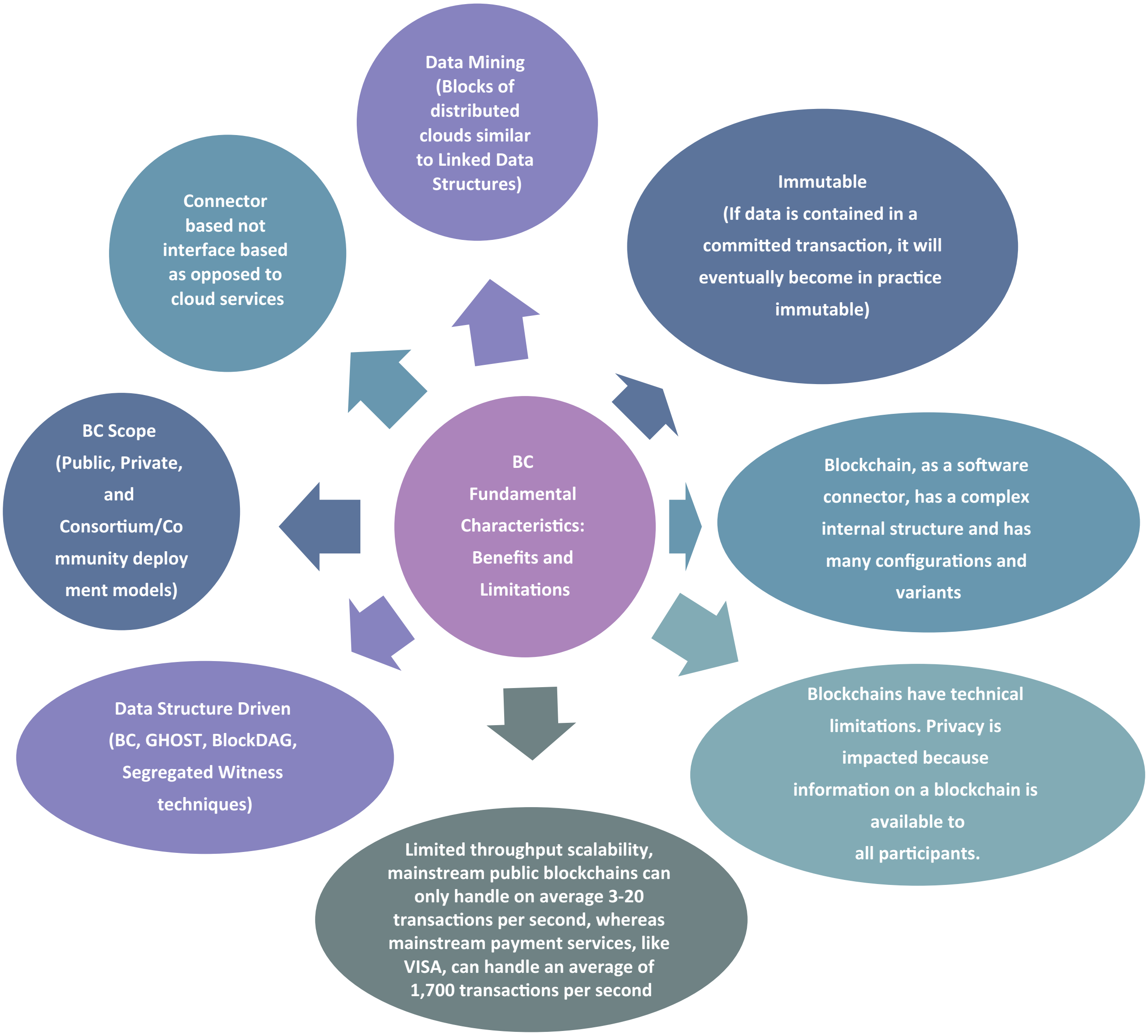

Data Mining (Blocks of distributed clouds similar to Linked Data Structures)
Connector based not interface based as opposed to cloud services
Immutable (If data is contained in a committed transaction, it will eventually become in practice immutable)
BC Scope (Public, Private, and Consortium/Co mmunity deploy ment models)
BC Fundamental Characteristics: Benefits and Limitations
Blockchain, as a software connector, has a complex internal structure and has many configurations and variants
Data Structure Driven (BC, GHOST, BlockDAG, Segregated Witness techniques)
Blockchains have technical limitations. Privacy is impacted because information on a blockchain is available to all participants.
Limited throughput scalability, mainstream public blockchains can only handle on average 3-20 transactions per second, whereas mainstream payment services, like VISA, can handle an average of 1,700 transactions per second

Fig. 3. Characteristics of blockchain applications requirements.

have proposed how software engineering can benefit from blockchain in terms of achieving continuous integration and packaging mechanism as such apt-get. In addition, they claim that blockchain can democratize and professionalize the SE profession ethically and legally, which can improve the quality of SE artifacts and increase the trust in software repositories and open-source software. Vacca et al.[17] proposed how SE can help the non-standard development of smart contracts. The framework shown in Figure 4 is known as S^{3}EF-HBCAs: Secure and Sustainable Software Engineering Framework for Healthcare Blockchain Applications, which provides best practices on full life-cycle support for the development of dApps:

- The secure requirements engineering method is known as BC-SQUARE, a systematic approach and steps in identifying, verifying, and validating blockchain applications using user stories and business process modeling and simulation (BPMN).
- The design method involves identifying and developing blockchain service components (BSC) and design assets with the Build Security In (BSI) approach proposed by Ramachandran.[18]
- Reference Architecture for blockchain dApps (REF-for-BC) is one of the key stages in the framework to map design assets identified during the design stage into four architectural layers such as blockchain AI and IoT application and prediction layer at the top, followed by the application layer where healthcare and or application specific BSC and assets such as API, interfaces, security, privacy, and test specific smart contract services.

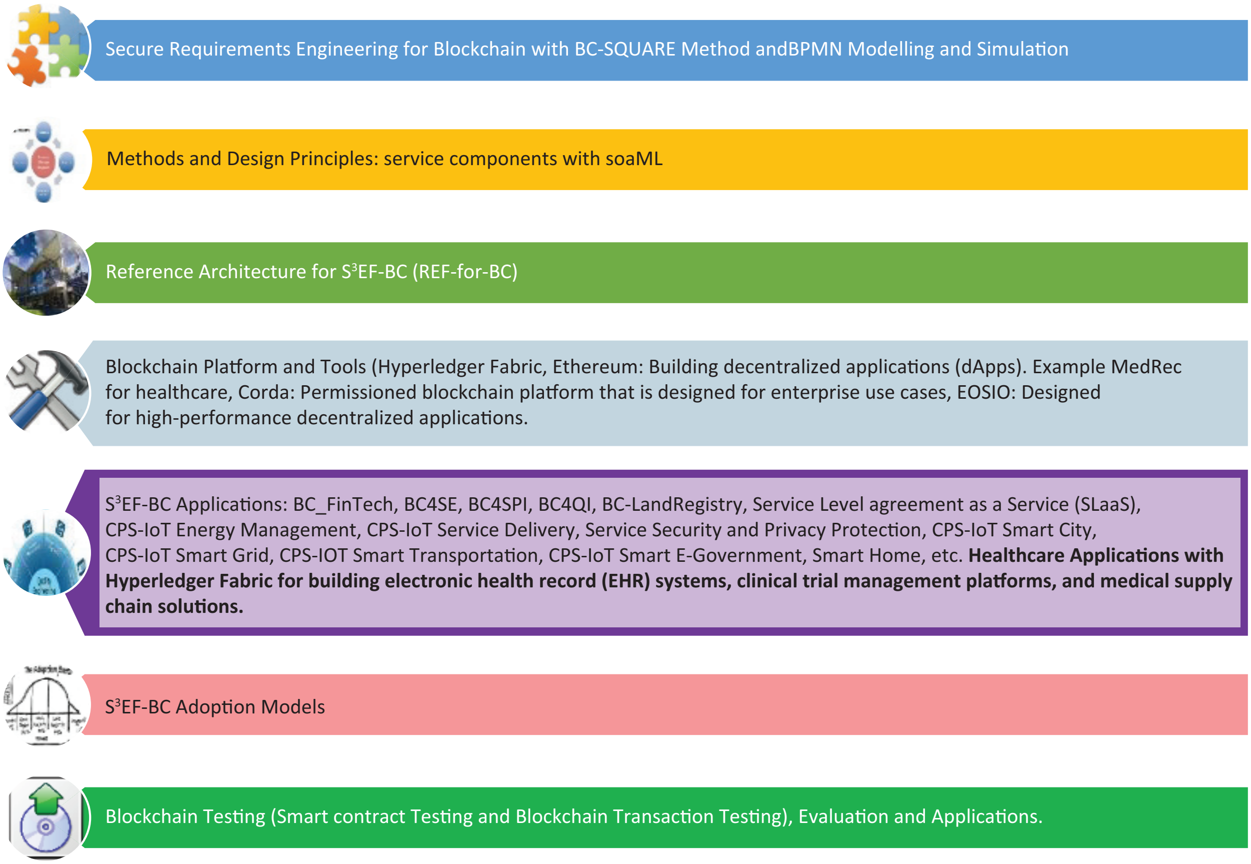

Secure Requirements Engineering for Blockchain with BC-SQUARE Method andBPMN Modelling and Simulation
Methods and Design Principles: service components with soaML
Reference Architecture for S3EF-BC (REF-for-BC)
Blockchain Platform and Tools (Hyperledger Fabric, Ethereum: Building decentralized applications (dApps). Example MedRec for healthcare, Corda: Permissioned blockchain platform that is designed for enterprise use cases, EOSIO: Designed for high-performance decentralized applications.
S3EF-BC Applications: BC_FinTech, BC4SE, BC4SPI, BC4QI, BC-LandRegistry, Service Level agreement as a Service (SLaaS), CPS-IoT Energy Management, CPS-IoT Service Delivery, Service Security and Privacy Protection, CPS-IoT Smart City, CPS-IoT Smart Grid, CPS-IOT Smart Transportation, CPS-IoT Smart E-Government, Smart Home, etc. **Healthcare Applications with Hyperledger Fabric for building electronic health record (EHR) systems, clinical trial management platforms, and medical supply chain solutions.**
S3EF-BC Adoption Models
Blockchain Testing (Smart contract Testing and Blockchain Transaction Testing), Evaluation and Applications.

Fig. 4. S^{3}EF-HBCAs: Secure and Sustainable Software Engineering Framework for Healthcare Blockchain Applications.

- Blockchain Platform and Tools (Hyperledger Fabric, Ethereum: Building decentralized applications (dApps). For example, MedRec for healthcare, Corda is a permissioned blockchain platform designed for enterprise use cases, and EOSIO is designed for high-performance decentralized applications.
- S^{3}EF-BC Applications: BC_FinTech, BC4SE, BC4SPI, BC4QI, BC-LandRegistry, Service Level Agreement as a Service (SLaaS), CPS-IoT Energy Management, CPS-IoT Service Delivery, Service Security and Privacy Protection, CPS-IoT Smart City, CPS-IoT Smart Grid, CPS-IOT Smart Transportation, CPS-IoT Smart E-Government, Smart Home, etc. Healthcare Applications with Hyperledger Fabric for building EHR systems, clinical trial management platforms, and medical supply chain solutions.
- S^{3}EF-BC Adoption Models. This involves creating awareness workshops, training, identifying strategic business innovation within the organization, ethical policies, and governance awareness on DAO.
- Blockchain Testing (Smart Contract Testing and Blockchain Transaction Testing), Evaluation, and Applications.

Adopting a systematic framework will help us achieve quality and standardization in establishing decentralized organizations with blockchain technology. The following section is devoted to presenting detailed processes and techniques of S^{3}EF-HBCA.

Requirements Engineering Method for Blockchain (Bc-Square)

Requirements Engineering plays a major role in any software development paradigm both in Agile SE as well as in traditional SE. Requirements engineering is a critical phase in software development that involves gathering, documenting, analyzing, and validating the needs and constraints of a system or application. It plays a pivotal role in ensuring that software meets the desired functionality, quality, and performance criteria. In the context of blockchain applications, requirements engineering is equally essential to guarantee the success of blockchain projects.

Blockchain technology, known for its decentralized and immutable nature, is being increasingly adopted in various industries such as finance, supply chain management, healthcare, and more. To harness the full potential of blockchain, it is essential to define and manage the specific requirements of blockchain applications effectively. In addition, there is also another major role blockchain technology plays in securing software development process and creating trust. As mentioned in the earlier section, there are two fields of research: *SE for blockchain, which adopts systematic processes and techniques for developing blockchain applications, and blockchain for SE,* which adopts the application of blockchain to improve the quality and security of software development process and assets. In requirements engineering, blockchain plays a major role in creating a smart contract between clients and software development organizations, as discussed by Dzhalila et al.[19]

A fundamental reference for understanding the importance of requirements engineering in developing blockchain applications is the paper titled 'A Systematic Literature Review on Blockchain Technology in Software Engineering'.[19] This paper highlights the challenges and best practices in SE for blockchain applications, emphasizing this technology's unique characteristics and considerations. In the realm of blockchain, requirements engineering involves:

- Defining Use Cases: Identifying and describing the specific use cases for the blockchain application, such as smart contracts for automated transactions or decentralized identity management.
- Security and Consensus Mechanisms: Determining the security requirements and consensus mechanisms to ensure the integrity and trustworthiness of the blockchain network.
- Scalability and Performance: Addressing the scalability and performance requirements to handle a potentially large number of transactions and participants.
- Interoperability: Ensuring interoperability with existing systems and other blockchain networks, if necessary.
- Privacy and Compliance: Meeting privacy and regulatory compliance requirements, especially in industries like finance and healthcare.
- User Experience: Designing user-friendly interfaces and experiences for interacting with the blockchain application.
- Testing and Validation: Developing test cases and validation criteria to ensure that the blockchain application meets its intended requirements.
- Maintenance and Evolution: Consider how the blockchain application will evolve and plan for maintenance and upgrades.

In summary, requirements engineering is a crucial step in developing blockchain applications. It helps ensure that blockchain technology is effectively leveraged to address specific business needs while considering the unique characteristics and challenges of blockchain networks. The development of requirements engineering for blockchain applications involves a structured approach to ensure the successful integration of security, privacy, and trust considerations. In addition, the requirements engineering for security methodology, known as Security Quality Requirements Engineering (SQUARE), has evolved at SEI (Software Engineering Institute).[20] This provides a framework for collecting basic security requirements early in the lifecycle. However, it lacks details and doesn't go beyond a nine-step process. Therefore, this paper proposes a requirements engineering method known as BC-SQUARE, as shown in Figure 5. BC-SQUARE method has been fine-tuned for identifying, collecting, validating by BPMN modeling and simulation, and evaluating quality blockchain quality requirements, which has been evaluated in a real-world case study presented in section 7 of this paper. This consists of several stages as follows:

- Agreed Definitions for Domain-Specific Blockchain Applications involve identifying common domain-specific terminology and definitions; therefore, they create an understandable communication amongst stakeholders.
- Identify Build Security, Privacy, and Trust In (BSPTI) Goals.
  - 'Build Security, Privacy, and Trust In' (BSPTI) goals are fundamental objectives in software development and system design that aim to incorporate security, privacy, and trust considerations into the development process from the outset. These goals are essential for creating robust and reliable systems that protect user data, maintain system integrity, and foster trust among users and stakeholders.

Here are some key BSPTI goals.

1. Security Goals:
   - Data Protection: Ensure the confidentiality and integrity of sensitive data by implementing encryption, access controls, and secure storage mechanisms.
   - Authentication and Authorization: Implement strong authentication methods and granular authorization controls to prevent unauthorized resource access.
   - Vulnerability Mitigation: Identify and address potential security vulnerabilities through techniques such as code reviews, penetration testing, and security scanning.
   - Incident Response: Develop a robust incident response plan to effectively detect and respond to security breaches.
   - Secure Coding Practices: Enforce secure coding standards and best practices to minimize the introduction of security flaws during development.

- Privacy Goals:
  - Data Minimization: Collect and process only the minimum amount of personal data necessary for the intended purpose.

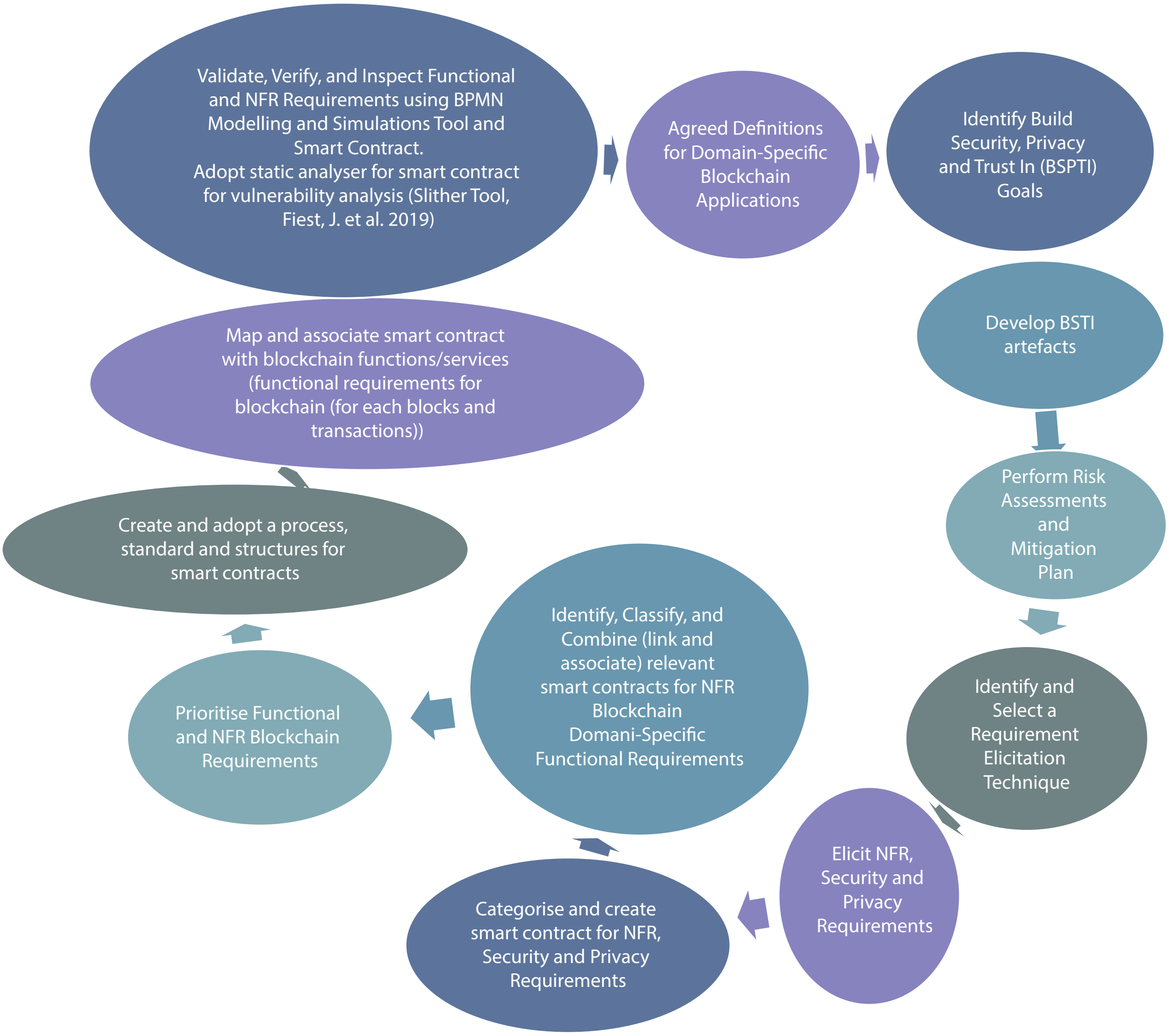

Validate, Verify, and Inspect Functional and NFR Requirements using BPMN Modelling and Simulations Tool and Smart Contract. Adopt static analyser for smart contract for vulnerability analysis (Slither Tool, Fiest, J. et al. 2019)
Agreed Definitions for Domain-Specific Blockchain Applications
Identify Build Security, Privacy and Trust In (BSPTI) Goals
Develop BSTI artefacts
Perform Risk Assessments and Mitigation Plan
Identify and Select a Requirement Elicitation Technique
Elicit NFR, Security and Privacy Requirements
Categorise and create smart contract for NFR, Security and Privacy Requirements
Identify, Classify, and Combine (link and associate) relevant smart contracts for NFR Blockchain Domani-Specific Functional Requirements
Prioritise Functional and NFR Blockchain Requirements
Create and adopt a process, standard and structures for smart contracts
Map and associate smart contract with blockchain functions/services (functional requirements for blockchain (for each blocks and transactions))

Fig. 5. Requirements engineering method for blockchain (BC-SQUARE) applications.

  - Consent Management: Obtain informed consent from individuals before collecting and processing their personal information.
  - Data Transparency: Provide users with clear and understandable privacy policies and notices regarding data collection and processing.
  - Data Portability: Enable users to access and transfer their data between services easily.
  - Data Retention: Define data retention policies and ensure the secure deletion of data when it is no longer needed.

- Trust Goals:
  - User Trustworthiness: Build systems that users can trust by being transparent about data use and adhering to ethical and legal standards.
  - Reliability and Availability: Ensure the system is highly available, reliable, and resilient to minimize downtime and disruptions.
  - Compliance: Adhere to industry-specific regulations and standards, such as GDPR, HIPAA, or ISO 27001, to demonstrate a commitment to compliance.
  - Third-Party Trust: Assess and verify the trustworthiness of third-party components, libraries, and services integrated into the system.
  - User Education: Educate users and stakeholders about security and privacy practices, empowering them to make informed decisions.

These BSPTI goals should be integrated into the software development lifecycle, from requirements gathering to design, coding, testing, deployment, and ongoing maintenance. Incorporating security, privacy, and trust considerations from the beginning helps reduce the risk of vulnerabilities and data breaches and fosters user confidence in the system.

- Develop Build Security, Privacy, and Trust In (BSTI) artifacts. This involves beginning by creating artifacts that document your 'Build Security, Privacy, and Trust In' (BSTI) goals, outlining your objectives for security, privacy, and trust within the blockchain application.
- Perform a risk assessment and mitigation plan involving a comprehensive risk assessment to identify potential threats and vulnerabilities related to your blockchain application. Develop a mitigation plan to address these risks.
- Identify and select a Requirement Elicitation Technique involves choosing appropriate requirement elicitation techniques (e.g. interviews, surveys, workshops) to gather requirements effectively from stakeholders and users.
- Elicit Non-Functions Requirements (NFR) for blockchain applications based on the characteristics identified in section 3, Trust, Security, and Privacy Requirements involves eliciting NFRs, trust, security requirements, and privacy requirements specific to your blockchain application. These requirements should align with your BSTI goals.
- Categorizing and creating smart contracts for NFR, Trust, Security, and Privacy Requirements involves translating the elicited NFRs, trust security, and privacy requirements into service-level agreements (SLAs) and smart contracts that will enforce these constraints on the blockchain.
- Identifying, classifying, and combining (linking and associating) relevant smart contracts for NFR Blockchain Domani-Specific Functional Requirements involves identifying and classifying the smart contracts that address specific functional requirements within the blockchain domain. Combine or link smart contracts as needed to meet functional requirements.
- Prioritizing Functional and NFR Blockchain Requirements involves prioritizing functional and non-functional requirements to allocate resources effectively and ensure that the most critical requirements are addressed first.
- Creating and adopting a process, standard, and structures for smart contracts based on SLA elicitations, SLA analysis, and SLA specification consists of defining the structure and logic for the smart contracts based on the identified requirements, ensuring that they are designed to meet trust, security, and privacy goals.
- Map and associate smart contracts with blockchain functions/services (functional requirements for blockchain (for each block and transaction)) consists of linking and associating the smart contracts with the relevant blockchain functions and services, aligning them with specific transactions and blocks as required by your application.
- Validate, verify, and inspect Functional and NFR Requirements using BPMN Modelling and simulation tools. Smart Contract involves utilizing BPMN modeling and simulation tools to validate and verify the functional and NFR requirements. Ensure that the smart contracts align with the intended business strategies and processes.
- Adopting a static analyzer for smart contracts for vulnerability analysis (Feist et al.[21] Slither Tool) involves deploying static analysis tools like Slither, Fiest, or other suitable tools to conduct vulnerability analysis on your smart contracts. Identify and rectify security vulnerabilities and weaknesses.

This systematic approach ensures that your blockchain application not only meets its functional requirements but also incorporates essential security, privacy, and trust elements from the early stages of development. It helps mitigate risks and enhances the reliability and trustworthiness of your blockchain application. Furthermore, according to Singh and Lee,[22] there needs to be more standards and a systematic process that follows RE principles to model smart contracts for Blockchain-Based Cloud (BBC) systems. As a result, the development quality is not assured, and issues such as scalability, trade-off between block size and security, and privacy leakage plague the development of smart contracts for BBC applications. Therefore, they have proposed a SLA-based RE for blockchain systems to identify and develop smart contracts: SLA Elicitation, SLA Analysis and Negotiation, SLA Specification, and SLA Assessment and Validation. However, their approach to developing smart contracts is very interesting. However, it lacks a development process for complete blockchain application requirements where we believe BC-SQUARE provides complete support for both functional and non-functional requirements and develops a reusable smart contract that aims to provide sustainability for blockchain technology as discussed in the introduction section. Figure 6 illustrates Requirements Engineering Classification for Blockchain Applications, consisting of reusable functional blockchain services such as domain-specific BCs and new generic BCs. The domain-specific BCs are further classified into a set of compute services and cloud-based BCs for secure data and transaction services. The newly developed generic BCs also support BC repository services for reuse and on-the-fly composition of blockchains.

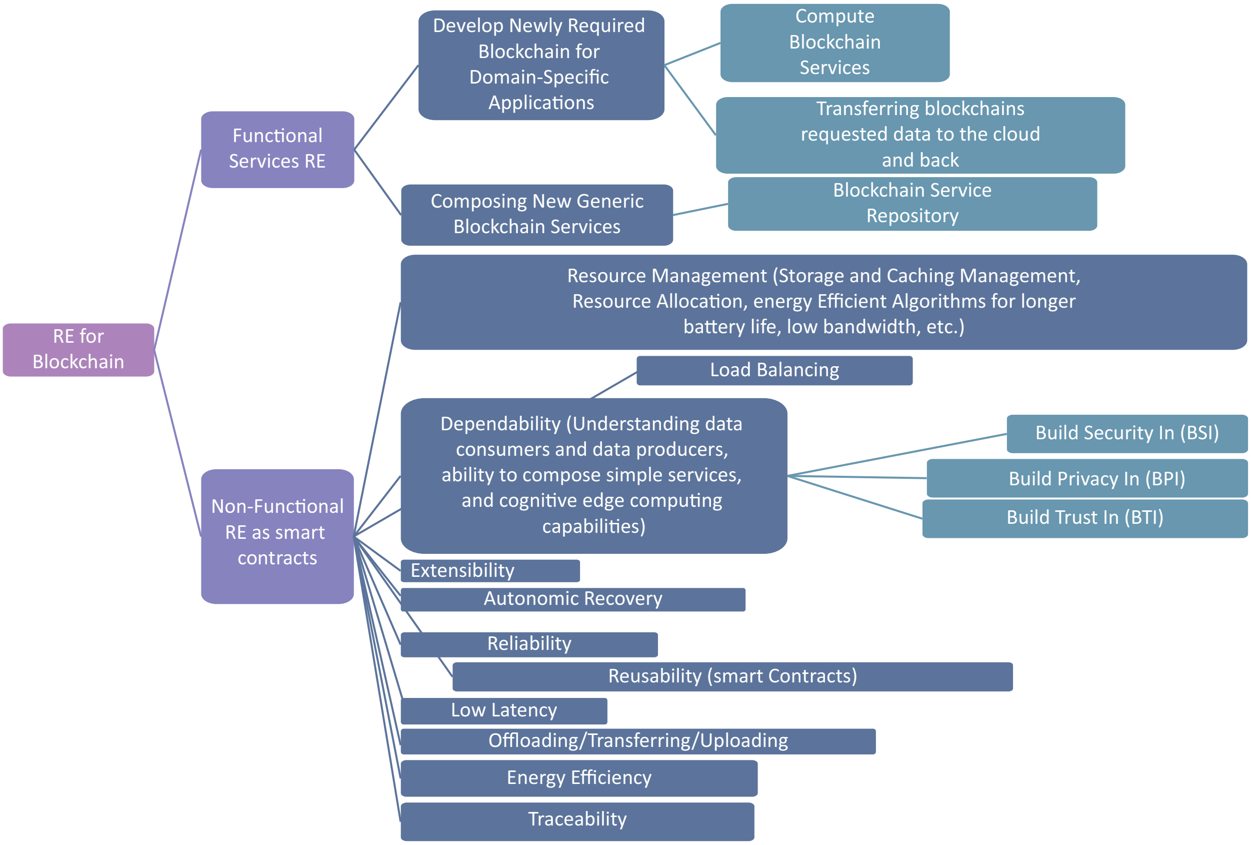

RE for Blockchain
Functional Services RE
Develop Newly Required Blockchain for Domain-Specific Applications
Compute Blockchain Services
Transferring blockchains requested data to the cloud and back
Composing New Generic Blockchain Services
Blockchain Service Repository
Non-Functional RE as smart contracts
Resource Management (Storage and Caching Management, Resource Allocation, energy Efficient Algorithms for longer battery life, low bandwidth, etc.)
Load Balancing
Dependability (Understanding data consumers and data producers, ability to compose simple services, and cognitive edge computing capabilities)
Build Security In (BSI)
Build Privacy In (BPI)
Build Trust In (BTI)
Extensibility
Autonomic Recovery
Reliability
Reusability (smart Contracts)
Low Latency
Offloading/Transferring/Uploading
Energy Efficiency
Traceability

Fig. 6. Requirements for engineering classification for blockchain applications.

Furthermore, the NFR smart contracts are classified into several NFR BCs, such as resource management BCs, Dependability BCs which are further classified into Build Security In (BSI) BCs, Build Privacy In (BPI) BCs, and Build Trust In (BTI) BCs. Furthermore, the classification of NFR BCs includes extensibility BCs, autonomous recovery, reliability, reusability, low latency, energy efficiency, offloading, and traceability. As discussed, there are two types of requirements known as functional requirements for blockchain (we can also call functional blocks) and non-functional requirements for blockchain, which are presented in the following sub-section.

Non-Functional Requirements for Blockchain as Reusable Smart Contracts

Designing NFRs as reusable smart contracts in blockchain applications offers several significant advantages and can greatly enhance the efficiency, security, and scalability of blockchain solutions. Here are the key reasons why designing NFRs as reusable smart contracts is important for standardization, consistency, efficiency, cost-effectiveness, sustainability, and productivity. Designing NFRs as reusable smart contracts in blockchain applications is a strategic approach that offers numerous benefits, including efficiency, security, customization, and compliance. It streamlines the development process, promotes best practices, and contributes to the scalability and trustworthiness of blockchain solutions.

Identifying, rationalizing, and addressing a set of NFRs for blockchain applications is essential because blockchain possesses unique characteristics and challenges that necessitate specific considerations. Here's a brief rationale for identifying NFRs for blockchain, focusing on key characteristics such as scalability, traceability, transparency, security, privacy, performance, consensus mechanisms, interoperability, legal and regulatory, resource efficiency, resilience, availability, and usability.

In summary, the characteristics of blockchain, such as decentralization, transparency, and immutability, bring about unique challenges and opportunities. Identifying and defining NFRs specific to these characteristics is crucial to ensure that blockchain applications are robust, secure, and capable of meeting the needs of various industries and use cases. It helps guide the development process and aligns the technology with the business objectives. Figure 7 illustrates the NFR for the medical supply chain. This kind of classifying NFR for domain-specific knowledge is useful for building reusable smart contracts.

One of the NFRs is sustainability which is becoming important for the sustainability of blockchain technology as well as addressing building reusable smart contracts, reproducible blockchains and energy-efficient algorithms, and optimization techniques in particular service transactions from cloud-based blockchains (CBB). The following section looks at how to granularize blockchains and how to identify reusable smart contracts and services using domain analysis methodologies, which include ontologies. This is useful for building reusable and sustainable dApps and establishing a product line approach.

Domain Classification for Blockchain Applications

Domain analysis in SE is a systematic process of studying and understanding a specific problem domain or application area to gather knowledge and insights that can inform the development of software systems within that domain. This process helps software engineers and developers create more effective and tailored solutions by identifying domain-specific requirements, constraints, and characteristics. It is often considered a critical initial step in software development when building systems for specialized or complex domains. Several studies on the application of domain analysis include consumer electronic products and blockchain in the supply chain.[24–30] Here are some key aspects of the domain analysis process in SE for the benefit of blockchain in business:

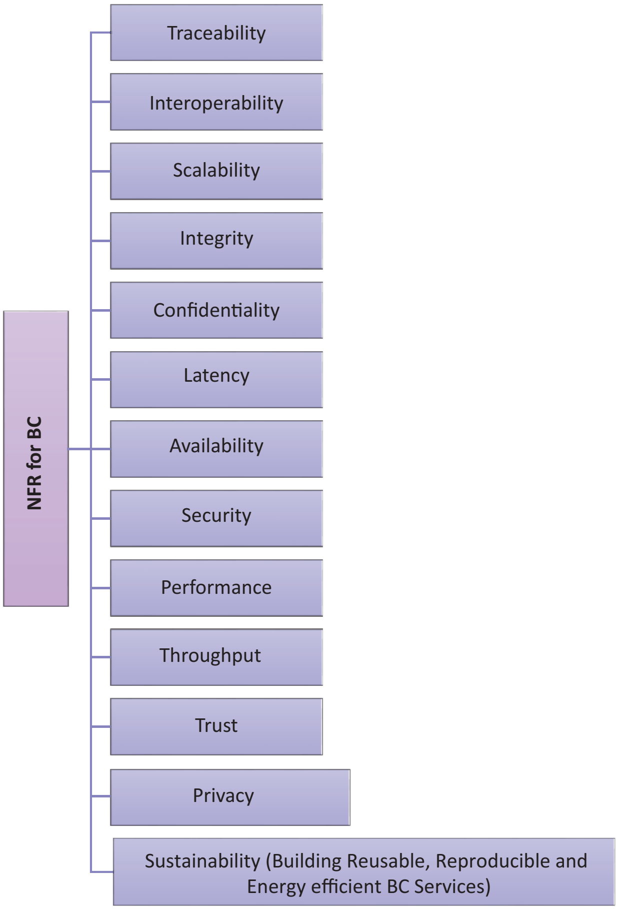


Fig. 7. Non-Functional requirements (NFR) for medical supply chain by Khatter and DevanjaliRelan.[23]

1. Understanding the Domain: This involves studying the domain's concepts, terminology, and underlying principles. It includes collaborating with domain experts to gain a deeper understanding of the problem space.
2. Understanding the Scope of the Domain to Develop dApps: This involves identifying the business scope and boundaries and will also help us build vertical and horizontal dApps product lines.
3. Identifying Requirements: Domain analysis helps in identifying domain-specific requirements and constraints that need to be addressed in the software system. These requirements may not be apparent without a thorough domain analysis.
4. Reusability: Domain analysis can lead to the identification of common patterns, components, or frameworks that can be reused across multiple projects within the same domain, promoting efficiency and consistency in software development.
5. Reducing Risks: By understanding the domain intricacies, software developers can make more informed decisions, reducing the risk of misunderstandings and costly errors during the development process.
6. Customization: Domain-specific knowledge obtained through analysis allows developers to customize the software to better meet the unique needs of the domain.
7. Documentation: Domain analysis often results in documentation that serves as a valuable reference for developers, stakeholders, and future maintainers of the software.
8. Validation: The findings of domain analysis can be used to validate and refine the software requirements, ensuring that the resulting system aligns with the domain's expectations.

Blockchain applications can be classified into different domains based on their intended use and functionality. Some common classes of application domain classification for blockchain applications include:

1. Financial Applications: These blockchain applications are primarily used in the financial sector, such as cryptocurrencies, digital wallets, and payment systems. They allow users to transfer funds securely and transparently without the need for intermediaries.
2. Supply Chain Management: Blockchain-based supply chain management applications are designed to provide greater visibility and accountability in the

movement of goods and services. They allow stakeholders to track the journey of a product from its origin to its final destination, ensuring the authenticity and quality of the product.

3. Healthcare: Blockchain applications in healthcare can improve the efficiency and security of medical data storage and sharing, ensuring patient privacy and data integrity. They can be used to create a secure and tamper-proof system for managing patient health records and facilitating medical research.
4. Identity and Access Management: Blockchain-based identity and access management applications can provide secure and decentralized verification of identity and access to services. They can be used for authentication and authorization of users, ensuring secure access to data and systems.
5. E-Government: Blockchain applications in government can improve the transparency, efficiency, and accountability of public services. They can be used for voting systems, public records management, and tax collection, among other use cases.
6. Smart Energy and Environment: Blockchain-based applications in the energy and environment sector can enable the tracking of renewable energy production and consumption, reducing carbon emissions and facilitating the trading of energy certificates.

Figure 8 shows a few examples of the different application domains for blockchain technology, and the list continues to grow as more use cases and adaptations of blockchain are discovered and developed.

There are several sub-domain and blockchain product lines and challenges that exist in healthcare applications, some of which include:

- Health Records Management: Blockchain can securely store and share EHRs among healthcare providers, ensuring data accuracy and patient consent.
- Drug Traceability: It can be used to trace the origin and distribution of pharmaceuticals, reducing counterfeit drugs and ensuring patient safety.
- Clinical Trials: Blockchain can improve transparency and integrity in clinical trials by recording and verifying trial data in a tamper-proof manner.
- Supply Chain Management: It enhances the tracking of medical supplies and devices, ensuring their authenticity and quality.
- Software Engineering Approach to DApps Development:
- Developing DApps in healthcare requires rigorous SE practices, including:
- Solid Codebase: Ensuring the DApp's codebase is robust, well-documented, and follows best practices to minimize vulnerabilities.

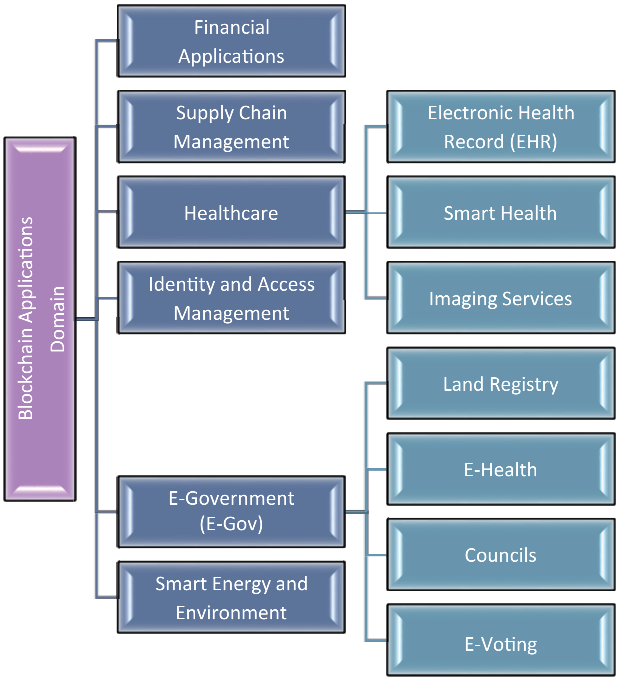


Fig. 8. Application domain classification for blockchain.

- Security Audits: Regular security audits and penetration testing to identify and mitigate vulnerabilities.
- Legal Compliance: Adhering to healthcare regulations like HIPAA (in the United States) to protect patient privacy and data security.
- User Experience: Creating a user-friendly interface for healthcare professionals and patients to interact with the DApp.
- Scalability: Choosing a blockchain platform that suits the specific healthcare use case and can scale as needed.
- Maintenance and Updates: Regular maintenance and updates to adapt to changing healthcare requirements and blockchain technology advancements.

In conclusion, blockchain technology has significant potential in healthcare, but its implementation requires careful consideration of security, privacy, scalability, and a strong SE approach. The success of blockchain applications in healthcare will depend on addressing these challenges and ensuring the technology's seamless integration with existing healthcare systems and regulatory frameworks.

Blocks-Points Effort Estimation

We also need to estimate the complexity of service level requirements and there have been several approaches such as the use case points (UCP) method, and user stories point estimation method in Agile Projects and they have also proposed a set of values for technical complexity factors (TCF) and environmental complexity factors depending on the nature of applications such as distributed computing, reusability, etc., and service point estimation method in SOA-based projects presented by.[31] This paper proposes a concept of block-point which involves identifying a sum of blocks or nodes in a blockchain contract and so forth, and recommends the use of BPMN modelling and simulation to model first-level requirements and to validate cost, resource, and performance constraints and smart contracts which can be reusable in the proposed SOA-based reference architecture discussed in the following section. In this context, this paper proposes a modified cloud COCOMO model[31] with weighting for cloud computing projects: a = 2, b = 2.1, c = 3, d = .2. Therefore, the effort and cost estimation equations are:

$$\textit{Blockchain project effort applied } (EA) = a \times (\textit{Block Points}) \; (\textit{Human Months}) \tag{1}$$

$$\textit{Blockchain development time } (dt) = c \times (\textit{Effort Applied})^{d} \; (\textit{Months}) \tag{2}$$

$$\textit{Number of Service Development Engineers Required} = \textit{Effort Applied } (EA) / \textit{Development Time } (dt) \tag{3}$$

Equations 1–3 provide cloud project effort and cost estimations based on process points which is the sum of all workflows (WF) divided by the total number of blockchain process activities (P).

$$\text{Number of Block Points} = \sum\nolimits_0^N WF \,/ \sum\nolimits_0^N P \;\; \text{X (TCF) X (ECF)} \tag{4}$$

TCF and ECF are useful factors for building sustainable blockchain services. Identifying TCF and ECF is crucial when building sustainable blockchain services. These factors help in assessing the challenges and intricacies involved in blockchain development within specific technical and environmental contexts. Here, we'll discuss the importance of TCF and ECF and provide citations to support the discussion.

Technical Complexity Factors (TCF)

Technical Complexity Factors (TCF) impact several technical factors on blockchain such as scalability, security, organizational impact of DAO, legal aspects of smart contracts and DAO, interoperability, and consensus mechanisms. These factors are critically analyzed as follows:

1. Scalability: TCFs related to scalability are essential for building sustainable blockchain services. As the blockchain network grows, it must handle an increasing number of transactions efficiently. Identifying scalability challenges and solutions ensures that the blockchain can meet future demands.[32]
2. Security: Blockchain's reputation is built on its security features. TCFs help identify potential vulnerabilities and security threats, enabling developers to implement robust security measures and protect the integrity of the blockchain. There are major concerns related to the security and legal aspects of blockchain, smart contracts, and organizational impact on DAO.[33,34,35]
3. Interoperability: The ability of a blockchain to interact with other systems and blockchains is vital for its sustainability. Identifying TCFs related to interoperability ensures that the blockchain can collaborate seamlessly with other technologies.
4. Consensus Mechanisms: TCFs associated with consensus algorithms impact the blockchain's efficiency and governance. Understanding the complexities of different consensus mechanisms helps in selecting the most suitable one for a particular use case.

Environment Complexity Factors (ECF)

Environmental complexity factors (ECF) refer to the external factors, conditions, and dynamics in a given environment or context that can significantly impact or influence a system, project, or organization. These factors encompass a wide range of elements, such as regulatory conditions, market dynamics, competitive landscape, technological advancements, user behavior, and more. ECFs are essential to consider when making decisions, formulating strategies, or assessing the sustainability and success of initiatives because they provide insights into the challenges and opportunities posed by the external environment. Understanding ECFs allows organizations to adapt and respond effectively to changes and uncertainties in their operating environment. The following are examples of ECFs that impact blockchain technology, and therefore it is essential to consider them earlier in the life cycle.

1. Regulatory Environment: ECFs related to regulatory compliance are critical for blockchain sustainability. Different regions have varying regulations, and understanding them is essential to avoid legal issues.
2. Market Dynamics and Market Competitiveness: ECFs tied to market conditions and competition impact the adoption and sustainability of blockchain services. Being aware of market trends and competitors' strategies helps in positioning blockchain solutions effectively.
3. User Adoption: ECFs related to user acceptance and adoption are essential. Identifying factors that may

hinder or promote user engagement is crucial for building a sustainable user base.

4. Technological Advances: ECFs associated with technological advancements can impact the longevity of blockchain services. Staying updated on emerging technologies ensures that blockchain solutions remain relevant and competitive.

In conclusion, identifying TCF and ECF is essential for building sustainable blockchain services. These factors help in addressing technical challenges, complying with regulations, staying competitive, and ensuring long-term relevance in a rapidly evolving blockchain landscape. By considering TCFs and ECFs, blockchain developers and organizations can make informed decisions that contribute to the success and sustainability of their blockchain initiatives. As a systematic framework to standardize the development of blockchain applications, the next phase in the framework is to adopt a systematic design process as the focus of the following section.

Design Strategies, Method, Dapp Development Process, Reference Architecture for Blockchain Applications (Ref-Arcbc)

It is important to understand dApps for Blockchain Application Development. Decentralized applications, commonly known as dApps, have revolutionized the world of blockchain technology and application development. In this brief introduction, we will define what a dApp is, clarify what it is not, and outline a systematic process for dApp development. A dApp, short for decentralized application, is a software application that operates on a blockchain network. Unlike traditional centralized applications that rely on a single central server, dApps leverage the principles of blockchain technology to run on a decentralized network of computers known as nodes. Key characteristics of dApps include:

- Decentralization: dApps operate on a peer-to-peer network of nodes, eliminating the need for a central authority or intermediary.
- Transparency: Transactions and data within dApps are recorded on a public ledger (blockchain), making them transparent and immutable.
- Security: The cryptographic nature of blockchain technology ensures the security and integrity of dApps, reducing the risk of fraud or data tampering.
- Open Source: Many dApps are open-source projects, encouraging community involvement and contributions.

Developing a dApp involves several key steps, and Ethereum development resource provides more detailed platform and learning resources (https://ethereum.org/en/developers/):

1. Idea and Conceptualization: Start by defining the problem your dApp will solve or the unique value it will provide to users. Consider the blockchain platform (e.g. Ethereum, Binance Smart Chain) that aligns with your goals.
2. Design and Architecture: Plan the user interface (UI), user experience (UX), and overall architecture of your dApp. Choose the appropriate blockchain and smart contract platform.
3. Smart Contract Development: Write and test the smart contracts that will power your dApp's functionality. Ensure security and efficiency in your code.
4. Front-End Development: Develop the front-end interface of your dApp, which interacts with the blockchain through web3 libraries or APIs.
5. Testing: Thoroughly test your dApp for functionality, security, and performance. Use testnets to simulate blockchain environments without real assets.
6. Deployment: Deploy your smart contracts to the chosen blockchain network and make your dApp accessible to users.
7. User Onboarding: Provide clear instructions for users to interact with your dApp, including wallet setup and transaction processes.
8. Community Engagement: Foster a community around your dApp to gather feedback, address issues, and encourage adoption.
9. Maintenance and Updates: Continuously monitor and maintain your dApp, addressing bugs, optimizing performance, and implementing updates.
10. Scaling: Explore options for scaling your dApp as user demand grows, considering solutions like layer 2 scaling or sidechains.

It is important to choose a blockchain dApp development process, design method, and data structure based on strategic business applications. Blockchain (dApps) design consists of several steps such as elaborating and mapping requirements, design rationale, and carefully choosing blockchain algorithms for more energy efficiency, computational efficiency, scalability, etc. as shown in Figure 9. This provides a development process for dApps which consists of several stages:

- Elaborating on dApp Requirements with BC-SQUARE involves gathering blockchain application requirements using the domain analysis method discussed in 5.2 and understanding stakeholder needs and business viability analysis.
- Creating Proof of Concept (POC) involves creating proof of concept by modeling and simulating business

requirements using Business Process Modelling and simulation tools such as BonitaSoft, Bizaghi, etc.

- Selecting your dApp platform involves Designing a rationale for selecting a decentralized application (dApp) development platform is a crucial step in the development process. Your choice of platform will significantly impact the functionality, scalability, security, and overall success of your dApp. For example, blockchain platforms in the healthcare domain include Ethereum which is one of the most popular blockchain platforms for dApp development, and it has been used for various healthcare applications such as patient records management, drug traceability, and telemedicine. Other platforms include Hyperledger Fabric, Corda, and MediBloc.
- Choosing a Blockchain Data Structure involves choosing the right blockchain data structures, which is a critical aspect of designing and building a blockchain system. The choice of data structures can significantly impact the blockchain's efficiency, security, and functionality.
- Design and Implement Blockchain and Mapping onto Reference Architecture involves Designing a blockchain-as-a-service (BaaS) platform involves creating a set of components and services that make it easier for developers and organizations to implement and use blockchain technology. Adopt a design strategy for creating a BaaS platform, implementing blockchain technology, and mapping it onto a reference architecture.
- Developing smart contracts involves Developing smart contracts systematically involves a structured approach to ensure the reliability, security, and efficiency of your blockchain-based applications. Adopt a systematic process for developing smart contracts such as defining the user requirements with use cases and BPMN as presented earlier, specifying smart contracts, choosing implementation platforms, and validating and verifying smart contracts.
- Choosing a front-end framework involves selecting the right front-end framework for blockchain applications, which is a crucial decision that impacts UX, development efficiency, and the overall success of your project. By following a specific design rationale, you should make an informed decision when selecting a front-end framework for your blockchain application, ensuring that it aligns with your project's requirements, development capabilities, and user expectations.
- Starting the testing cycle involves testing blockchain decentralized applications (dApps) requires a systematic approach and specialized test techniques due to the unique characteristics of blockchain technology. Adopt a systematic process and test techniques to start the testing cycles for blockchain dApps such as defining test objectives, creating test plans, and test environments, and creating test cases, test techniques including unit testing, functional testing, integration testing, and energy-efficiency testing.

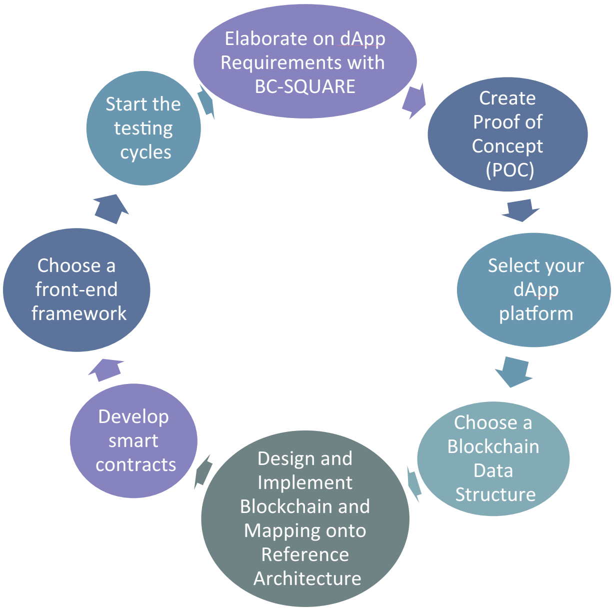

Elaborate on dApp Requirements with BC-SQUARE
Create Proof of Concept (POC)
Select your dApp platform
Choose a Blockchain Data Structure
Design and Implement Blockchain and Mapping onto Reference Architecture
Develop smart contracts
Choose a front-end framework
Start the testing cycles

Fig. 9. Design and development process for dApps.

Data structures play a critical role in various computer science and distributed systems, and when comparing and evaluating them, it's essential to consider their design, purpose, and performance characteristics. In this response, I'll compare and critically evaluate several data structures, including Blockchain, GHOST (Greedy Heaviest Observed Subtree), BlockDAG (Directed Acyclic Graph), and Segregated Witness, in terms of their design, use cases, and notable features. Table 1 on Blockchain Data Structures provides a list of data structures such as Blockchain, GHOST, BlockDAG, and Segregated witness.

Critical Evaluation of Blockchain Data Structures

Critical evaluation of the various blockchain data structures should be based on non-functional requirements identified for your business strategy such as security, scalability, use cases, complexity, and adoption as follows:

- Scalability: BlockDAGs and SegWit address scalability challenges more effectively than traditional blockchains like Bitcoin. They enable higher throughput and lower transaction costs.
- Security: Blockchain and GHOST provide strong security through their consensus mechanisms. SegWit enhances security by reducing the risk of transaction malleability attacks.

Table 1. Blockchain data structures

Blockchain Design Methods and Its Data Structures	Design Rationale	Use Cases	Notable Features
Blockchain	Blockchain is a distributed, append-only ledger that relies on a linear chain of blocks, where each block contains a list of transactions. Blocks are linked through cryptographic hashes, ensuring the integrity and immutability of the ledger.	Blockchain is popular in applications like cryptocurrencies (e.g. Bitcoin), supply chain management, and smart contracts (e.g. Ethereum).	Blockchain provides strong security through Proof of Work (PoW) or Proof of Stake (PoS) consensus mechanisms, but it can suffer from scalability and latency issues due to its linear structure.
GHOST (Greedy Heaviest Observed Subtree)	GHOST is a data structure used to resolve forks and reach consensus in Ethereum's blockchain. It considers not just the longest chain but also includes orphaned blocks, giving a more comprehensive view of the network.	GHOST is primarily used in blockchain networks that adopt Ethereum's protocol for consensus.	GHOST enhances security by considering more blocks in the consensus process but can lead to increased complexity in certain cases.
BlockDAG (Directed Acyclic Graph)	BlockDAG is a data structure that allows multiple blocks to reference one another in a directed acyclic graph rather than a linear chain. This structure eliminates the need for a single, global consensus point.	BlockDAG is used in cryptocurrencies like IOTA and Nano to address scalability and throughput issues associated with traditional blockchains.	BlockDAGs offer improved scalability and reduced confirmation times compared to traditional blockchains. However, they require different consensus algorithms, such as Tangle or DAG-based PoW.
Segregated Witness (SegWit)	SegWit is a data structure upgrade for Bitcoin. It separates transaction data and witness data, allowing for more efficient use of block space and fixing transaction malleability issues.	SegWit is specific to the Bitcoin network and aims to improve its scalability and security.	SegWit reduces the size of transactions, enabling more transactions to fit in a block. This helps reduce transaction fees and enhance the overall efficiency of the Bitcoin network.

- Complexity: GHOST introduces complexity by considering orphaned blocks. BlockDAGs are more complex to implement due to the lack of a linear chain.
- Use Cases: The choice between these data structures depends on the specific use case. BlockDAGs are suitable for high-throughput applications, while SegWit is tailored for Bitcoin's needs.
- Adoption: Blockchain and SegWit have seen widespread adoption, while GHOST and BlockDAGs are adopted in specific blockchain networks.

Marches et al.[36] propose several Agile practices based on user stories and UML modeling. They also argue Agile methods are well suited for dApps development as they offer for self-organized teams and requirements are not well understood initially. They also proposed other Agile techniques such as Continuous Testing, Test Driven Design, Refactoring, Continuous Integration, Collective code ownership, Information Radiators (Cards, Boards, Burndown charts), Coding Standards, and Pair Programming (in some cases).

In conclusion, the choice of a data structure depends on the requirements and goals of a given blockchain or distributed ledger system. Each of these data structures has its advantages and disadvantages, and selecting the right one involves considering factors like scalability, security, complexity, and the intended use case. Furthermore, the landscape of blockchain technology is continually evolving, with innovations and data structures emerging regularly.

Component-Based Software Design for Blockchain

Component-Based Software Design for Blockchain is important as it provides a natural extension of service by providing required and provider services to other blocks in the blockchain. Component-based software design is crucial for blockchain development as it offers a systematic approach to building complex blockchain systems. This approach provides a natural extension of services, enabling required services to be provided to other blocks in the blockchain. Here are several reasons why component-based software design is essential for blockchain:

1. Modularity and Reusability: Component-based design promotes modularity by breaking down a blockchain system into smaller, self-contained components or smart contracts. These components can be reused in various parts of the blockchain or even in different blockchain applications. This reusability saves time and resources in development.
2. Service Composition of Blockchain: Blockchain systems can leverage a wide range of services, including oracles, data feeds, identity management, and more. Component-based design enables the composition of

these services, allowing for the creation of complex, feature-rich dApps (decentralized applications). An example of a blockchain service component is shown in Figure 10, which is a service component model for data analytics consisting of several provider services (shown as Lolli pop symbol) such as Interface on Data Pre-Processing (IDataPreProcessing), etc.

Blockchain Security

Blockchain has arisen as a potent security solution for various applications, primarily due to one of its fundamental attributes: immutability. Regrettably, Destefanis et al.[4] highlighted vulnerabilities found in smart contract libraries and the insecure programming of smart contracts. Consequently, there is a growing need for the advancement of blockchain SE to address these issues. Therefore, this paper proposes, as shown in Figure 11, a concept of security service smart contract component and architecture which is a reusable smart contract and can be plugged into across applications. This improves the security, scalability, and immutability of smart contracts.

The BSC model for security smart contracts as shown in Figure 11 provides several required interfaces such as ISignature for identity management, IEncryption for encryption for cryptographic algorithms, etc. The following section is devoted to presenting the reference architecture for dApp development for standardization and sustainability.

Reference Architecture for Blockchain (REF-ArcBC)

The integration of blockchain, AI, and IoT technologies can enable the development of powerful and innovative applications that can transform various industries. To standardize the development of AI-enabled blockchain applications and blockchain-driven AI applications requires a reference architecture. A reference architecture for blockchain applications is a standardized blueprint or framework that provides a structured and well-defined approach to designing and building blockchain-based solutions. It serves as a foundational guide for developers, architects, and organizations looking to leverage blockchain technology effectively. Here's a brief introduction to the importance and meaning of a reference architecture for blockchain applications:

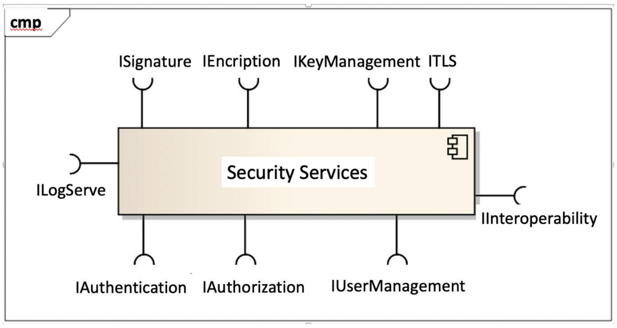


Fig. 11. Blockchain service component for security smart contract.

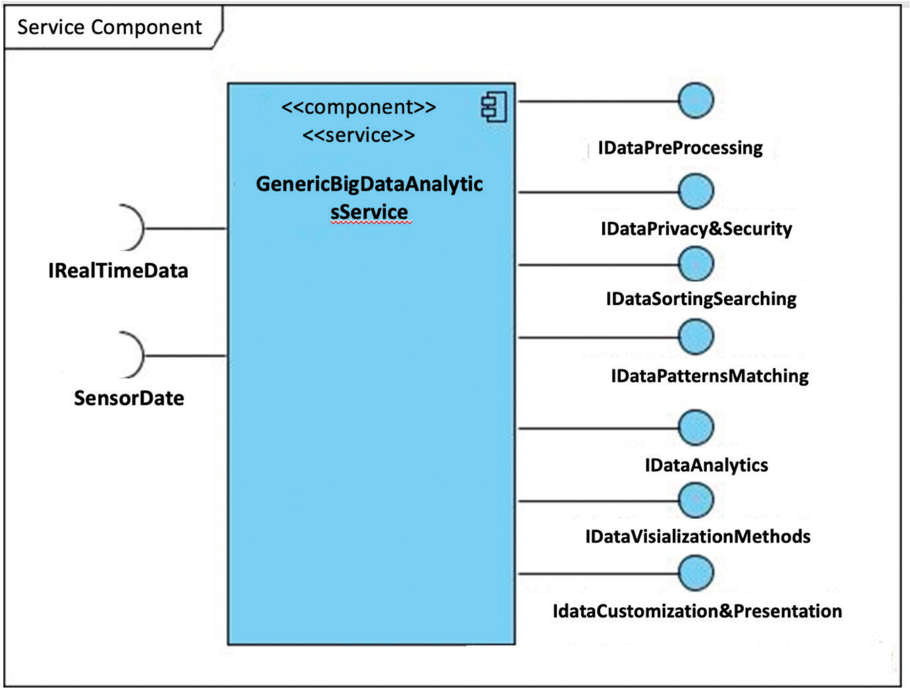


Fig. 10. Blockchain service component model.

1. Standardization and Consistency: A reference architecture establishes a common set of design principles, best practices, and components, ensuring consistency across different blockchain applications. This standardization streamlines development and maintenance processes.
2. Efficiency: It helps developers avoid reinventing the wheel. By following a reference architecture, they can leverage pre-established patterns and components, reducing development time and costs.
3. Interoperability: Reference architectures often consider interoperability with existing systems and other blockchain networks. This is vital for ensuring that blockchain applications can seamlessly work with other technologies.
4. Security: Security is a paramount concern in blockchain applications. A reference architecture typically incorporates security best practices and guidelines, which helps in reducing vulnerabilities and risks.
5. Scalability: As blockchain applications grow, scalability becomes a critical factor. A reference architecture guides how to design systems that can easily scale to accommodate increased loads.
6. Adoption and Collaboration: A reference architecture facilitates collaboration and adoption by providing a common framework that different organizations and developers can use. This leads to faster adoption of blockchain technology.
7. Regulatory Compliance: Compliance with regulatory requirements is a significant challenge in the blockchain space. A reference architecture may include guidelines on how to design systems that adhere to relevant regulations.
8. Flexibility: While offering a structured approach, a reference architecture is often flexible enough to accommodate variations based on specific use cases or industry requirements.

In summary, a reference architecture for blockchain applications is essential for establishing a standardized, efficient, and secure foundation for developing and implementing blockchain solutions. It ensures that blockchain technology is used consistently and effectively, promoting interoperability, security, and scalability, while also aiding in regulatory compliance and encouraging wider adoption.

The reference architecture for blockchain is illustrated in Figure 12 as Reference Architecture for Blockchain (REF-ArcBC). REF-ArcBC consists of four layers namely: BC AI and IoT Application and Prediction Layer, Application Layer, Blockchain Layer, and Infrastructure Layer. The application layer for such applications typically consists of the following components as shown in Figure 12.

- Blockchain Layer: This layer includes the blockchain network and infrastructure that stores and validates data. It ensures the immutability, security, and transparency of the data by using cryptographic algorithms, consensus mechanisms, and smart contracts.
- AI Layer: This layer includes machine learning algorithms, neural networks, and other AI models that enable the analysis and processing of large amounts of data. It can be used for various purposes, such as predictive analytics, anomaly detection, and natural language processing.
- IoT Layer: This layer includes the physical devices, sensors, and gateways that collect and transmit data to the blockchain and AI layers. It can enable real-time monitoring and control of various systems and processes, such as smart homes, smart cities, and industrial automation.
- Integration Layer: This layer includes the middleware and APIs that enable the integration and interoperability of the blockchain, AI, and IoT layers. It can ensure the seamless exchange of data and transactions between the different layers, enabling the development of complex and decentralized applications.
- Application Layer: This layer includes the UIs, dashboards, and other applications that enable users to interact with the blockchain, AI, and IoT layers. It can provide various features and functionalities, such as data visualization, decision-making support, and automation.

These layers can be combined and customized based on the specific requirements and use cases of the applications. The integration of blockchain, AI, and IoT technologies can enable the development of innovative solutions that can address various challenges and opportunities in different industries. These are just a few examples of the services provided by the blockchain application layer. The actual services and functionalities depend on the specific use cases and requirements of the applications. The blockchain layer in blockchain technology refers to the underlying network and infrastructure that stores and validates data using cryptographic algorithms, consensus mechanisms, and smart contracts. Some of the services that happen at the blockchain layer include decentralized data storage, data validation and consensus, smart contract execution, cryptographic security, tokenization, digital asset management, interoperability, and integration.

These are some of the services that happen at the blockchain layer in blockchain technology. The actual services and functionalities provided by the blockchain layer depend on the specific use cases and requirements of the applications. The infrastructure layer in a blockchain reference architecture provides the underlying technical infrastructure that supports the blockchain network and

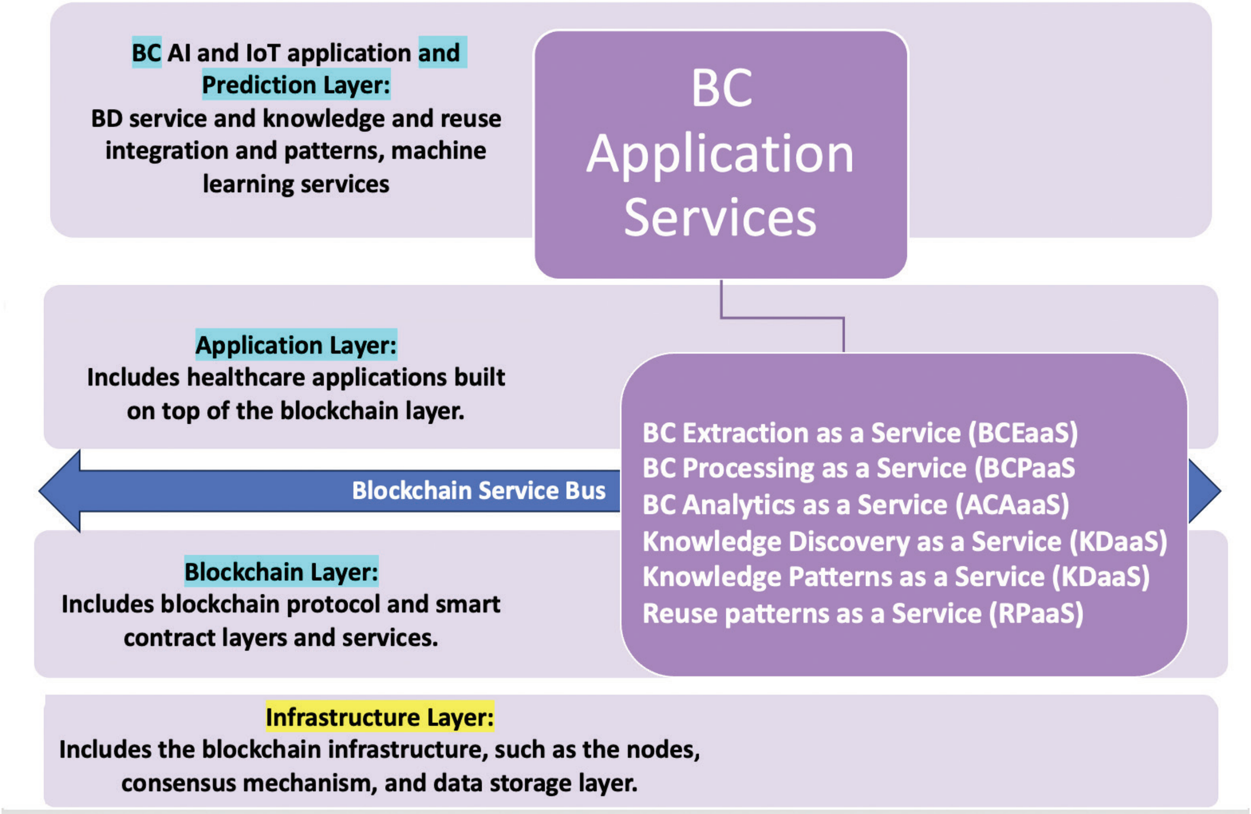

BC AI and IoT application and Prediction Layer:
BD service and knowledge and reuse integration and patterns, machine learning services
BC Application Services
Application Layer:
Includes healthcare applications built on top of the blockchain layer.
Blockchain Service Bus
BC Extraction as a Service (BCEaaS)
BC Processing as a Service (BCPaaS
BC Analytics as a Service (ACAaaS)
Knowledge Discovery as a Service (KDaaS)
Knowledge Patterns as a Service (KDaaS)
Reuse patterns as a Service (RPaaS)
Blockchain Layer:
Includes blockchain protocol and smart contract layers and services.
Infrastructure Layer:
Includes the blockchain infrastructure, such as the nodes, consensus mechanism, and data storage layer.

Fig. 12. Reference architecture for blockchain (REF-ArcBC).

enables the execution of smart contracts and other decentralized applications. The services provided by the infrastructure layer include:

- Node Management: The infrastructure layer provides services for the management of the nodes that participate in the blockchain network. These services include node registration, node discovery, node synchronization, and node communication.
- Network Consensus: The infrastructure layer provides services for the consensus mechanism that enables multiple nodes in the network to validate and agree on the data and transactions stored on the blockchain. These services include consensus algorithm implementation, block creation, block validation, and block propagation.
- Data Storage and Retrieval: The infrastructure layer provides services for the storage and retrieval of data on the blockchain network. These services include block storage, transaction storage, and data retrieval through APIs.
- Smart Contract Execution: The infrastructure layer provides services for the execution of smart contracts and other decentralized applications on the blockchain network. These services include smart contract development, deployment, and execution.
- Security and Privacy: The infrastructure layer provides services for the security and privacy of the blockchain network and its participants. These services include cryptographic mechanisms, access control mechanisms, and identity management services.
- Interoperability and Integration: The infrastructure layer provides services for the interoperability and integration of the blockchain network with other networks and applications. These services include inter-chain communication, cross-chain transaction support, and application programming interfaces (APIs).
- Scalability and Performance: The infrastructure layer provides services for the scalability and performance of the blockchain network. These services include sharding, sidechains, and other techniques that enable the network to handle large volumes of data and transactions.

These are some of the services provided by the infrastructure layer in a blockchain reference architecture. The actual services and functionalities depend on the specific use cases and requirements of the blockchain application.

It's crucial to recognize that assessing intricate reference architectures like REF-ArcBC demands a multiyear effort to comprehensively gauge their performance in real-world scenarios and assess their adaptability for making global smart contract revisions. Consequently, the subsequent sections delineate a method for evaluating REF-ArcBC, which centers on a real-world case study concerning a chatbot, an application of conversational AI. This evaluation approach encompasses the utilization of BPMN tools, specifically leveraging Bizaghi for modeling and simulation.

Case Study and Evaluation of Ref-Arcbc with Bpmn: Electronic Health Record

An EHR is a digitalized, longitudinal record of a patient's health and medical history. It includes information about a patient's medical conditions, treatments, medications, allergies, laboratory results, and more. The EHR is designed to streamline healthcare processes, improve the quality of care, enhance patient safety, and facilitate the sharing of information among healthcare providers. They have become a fundamental part of modern healthcare systems, enabling healthcare professionals to access and update patient records securely and efficiently. EHR[37] and Ekblaw et al.[38] describe the benefits of HER and its features. A list of smart contracts that can be used for building EHR systems using the Ethereum programming language are as follows:

- Patient Registry Smart Contract: This contract can be used to store patient demographic information such as name, date of birth, gender, and contact information. It can also be used to store other relevant information, such as medical history and allergies.
- Electronic Health Record Smart Contract: This contract can be used to store patient health information such as diagnoses, medications, laboratory test results, and imaging studies. It can also be used to track changes to the patient's health status over time.
- Consent Smart Contract: This contract can be used to manage patient consent for the use and sharing of their health information. It can be used to record the patient's consent preferences and to manage access to their health information by healthcare providers and other authorized parties.
- Identity Smart Contract: This contract can be used to manage patient identity and authentication. It can be used to verify the patient's identity and to ensure that only authorized users have access to their health information.
- Payment Smart Contract: This contract can be used to manage payments for healthcare services. It can be used to automatically process payments for services rendered and to manage disputes between patients and healthcare providers.
- Prescription Smart Contract: This contract can be used to manage the prescription of medications. It can be used to track prescriptions, monitor adherence to medication regimens, and manage refills.
- Medical Device Smart Contract: This contract can be used to manage the use and maintenance of medical devices. It can be used to track device usage, monitor device performance, and manage device maintenance and repairs.

These are just a few examples of smart contracts that can be used for building EHR systems using the Ethereum programming language. Smart contracts can be customized and combined in various ways to meet the specific needs of healthcare organizations and patients.

Business Process Modelling (BPMN), Simulation, and Evaluation can be applied to blockchain applications development (dApps) by creating a model scenario representation of a business process or system. BPMN allows for the visualization and documentation of the various steps, activities, and decisions involved in a process. This information can then be used to create a blockchain scenario, which is a simulation replica of the physical system or process. The blockchain application requirements can be modeled, simulated, and evaluated to analyze its performance, identify bottlenecks, and optimize resource allocation and business processes. There are numerous BPMN modeling and simulation tools exist including BonitaSoft, Bizagi, etc. Chang et al.[39] provide a critical evaluation of the BPMN tools. In addition, Gao et al. have applied BPMN to the implementation of an enterprise resource planning (ERP) system in manufacturing. Therefore, we believe BPMN is extremely valuable to blockchain application modeling and simulation.

By using BPMN modeling, simulation, and evaluation, organizations can gain a deeper understanding of their processes and systems. They can simulate different scenarios, test out various changes and improvements, and evaluate the impact on resource utilization and business performance. This helps in identifying inefficiencies, optimizing resource allocation, and improving overall productivity.

For example, a manufacturing company can create dApps of their production line using BPMN. By simulating and evaluating different production scenarios, they can identify the optimal allocation of resources such as machines, manpower, and materials. They can analyze the impact of process changes on productivity, throughput, and quality. This enables them to optimize their resources, reduce costs, and streamline their business processes. Figure 13 shows an example of how BPMN has been modeled representing the reference architecture for blockchain presented in the earlier section.

Business Process Model and Notation (BPMN) is a widely used standard for representing and designing business processes in a visual, standardized format. Here's a brief introduction to BPMN notation, modeling, processes, and simulation steps:

1. BPMN Notation: BPMN provides a set of symbols and conventions for visually representing business processes. It includes various elements such as tasks, gateways, events, flows, and pools, each with specific meanings and functions. These symbols are used to create diagrams that illustrate the flow and structure of a business process.
2. Modeling: BPMN allows businesses to create visual models of their processes, making it easier to understand, document, and analyze complex workflows. BPMN diagrams are typically divided into various elements, including start and end events, activities (tasks), gateways (decision points), and connecting flows that depict the sequence of actions.
3. Processes: A BPMN process represents a series of connected tasks and activities that together form a complete business process. These processes can range from simple, linear workflows to complex, branching, and parallel processes, capturing the way an organization operates.
4. Simulation Steps: Simulation is a valuable aspect of BPMN modeling, as it enables businesses to assess and optimize their processes before implementation. The steps for simulating a BPMN process typically include:
    - Model Creation: Begin by creating a BPMN diagram that represents the desired process. This includes defining tasks, decision points, and the flow of activities.
    - Data and Resource Allocation: Specify data inputs and outputs for each activity and allocate resources (people, equipment) as necessary.
    - Parameterization: Assign values to process parameters to evaluate various scenarios and assess their impact on the process.
    - Simulation Execution: Use BPMN simulation tools to run the model. This simulates the actual execution of the process, considering factors like task duration, resource availability, and decision outcomes.
    - Analysis and Optimization: Evaluate the simulation results to identify bottlenecks, inefficiencies, or areas for improvement. Adjust the model and parameters as needed to optimize the process.
    - Documentation and Reporting: Document the simulation results and findings. This information can be used to make informed decisions about process enhancements or automation.

Furthermore, the BPMN simulation provides validation and verification of the blockchain application requirements, and the reference architecture and we can replicate architectural layers as the swim lanes and sub-systems as pools. For this paper, we have used the Bizagi modeler and there are plenty of open-source tools

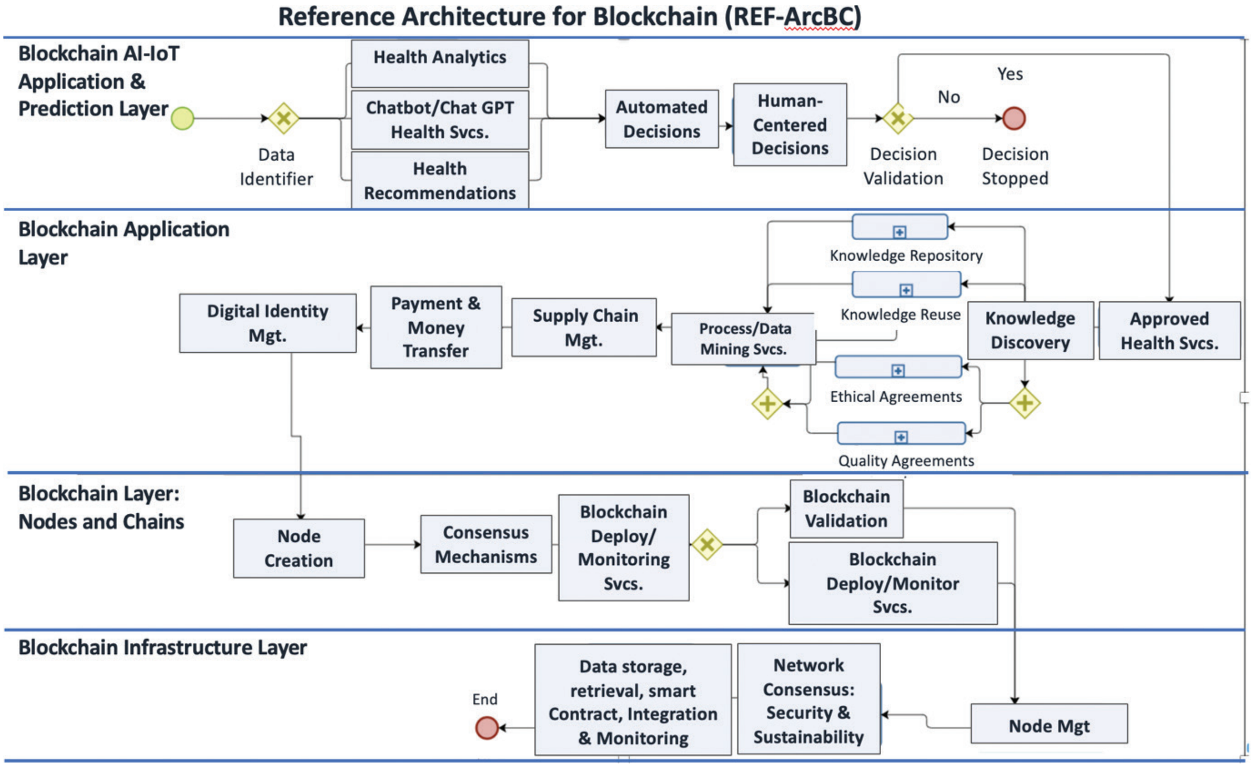

Reference Architecture for Blockchain (REF-ArcBC)
Blockchain AI-IoT Application & Prediction Layer
Data Identifier
Health Analytics
Chatbot/Chat GPT Health Svcs.
Health Recommendations
Automated Decisions
Human-Centered Decisions
Decision Validation
No
Yes
Decision Stopped
Blockchain Application Layer
Digital Identity Mgt.
Payment & Money Transfer
Supply Chain Mgt.
Process/Data Mining Svcs.
Knowledge Repository
Knowledge Reuse
Knowledge Discovery
Approved Health Svcs.
Ethical Agreements
Quality Agreements
Blockchain Layer: Nodes and Chains
Node Creation
Consensus Mechanisms
Blockchain Deploy/ Monitoring Svcs.
Blockchain Validation
Blockchain Deploy/Monitor Svcs.
Blockchain Infrastructure Layer
End
Data storage, retrieval, smart Contract, Integration & Monitoring
Network Consensus: Security & Sustainability
Node Mgt

Fig. 13. REF-Arc BPMN model for electronic healthcare record (Bizagi Modeller).

available in the marketplace. Chang et. al.[39] have provided a critical evaluation of BPMN tools.

Business Process and Reference Architecture Efficiency Estimation Metrics

The BPMN model shown in Figures 13 and 14 represents an example scenario for EHR health sensor data in real-time from which we can generalize the efficiency of the process execution mathematically for REF-ArcBC processes. The efficiency of the BPMN models can be estimated using several BPMN model parameters such as the total number of blockchain processes (P) + the total number of parallel blockchain processes (PP) + total number of decisions and gateways (G) + total number of swim lanes (SL). The efficiency of the REF-ArcBC is measured based on several key BPMN model metrics as follows:

REF-ArcBC Efficiency (REF-ArcBCE%) = (No. of BPMN blockchain processes (P) + No. of parallel BPMN blockchain process (PP)) + number of decisions and gateways (G) + total number of swim lanes (SL)/Total Number of BPMN Blockchain Processes (N)

$$REF - ArcBCE = \left(\sum_{1}^{N} P + \sum_{1}^{N} PP\right) + /\mathrm{N} \tag{5}$$

Business Process Efficiency (%) = (REF-ArcBC*Total Number of BPMN Blockchain Process Execution Time (BPMNt/Total No. of BPMN Blockchain Processes)*100

$$BPMNe = (REF - ArcBCE * BPMNt/N) \tag{6}$$

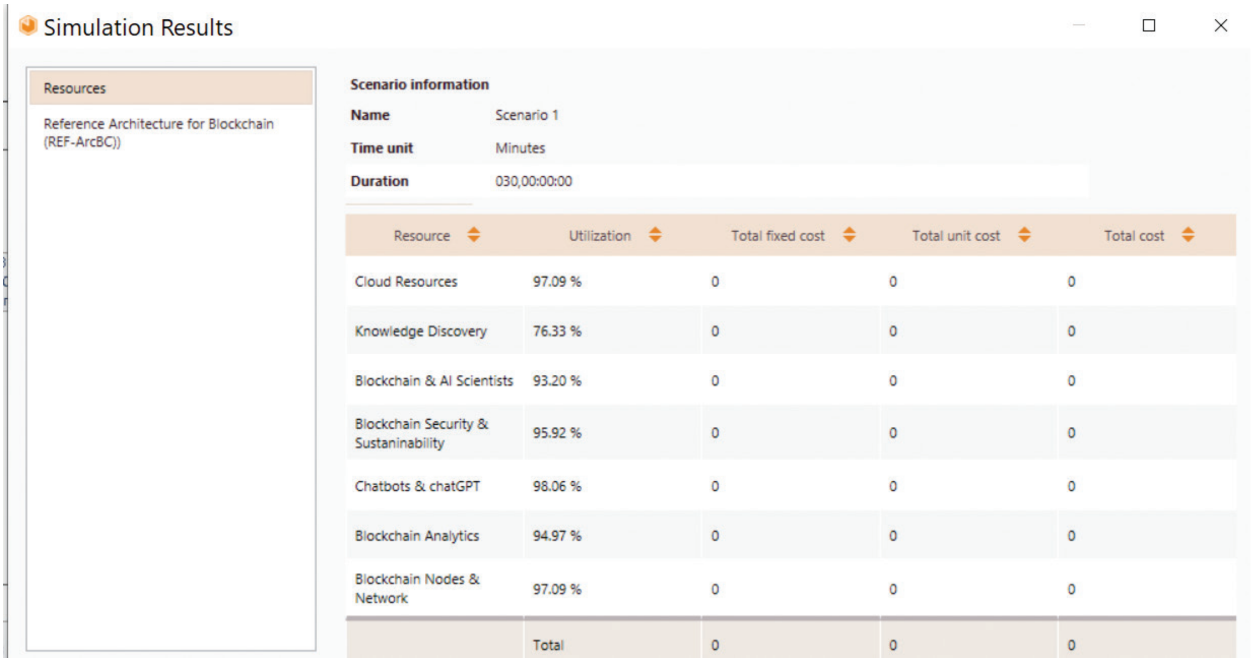
Equations (5) and (6) provide a measurable efficiency and sustainability of the REF-ArcBC and business process modeling.

In summary, BPMN notation provides a standardized way to visually represent and model business processes. Simulation of these models allows organizations to analyze and optimize their processes for increased efficiency and effectiveness, ultimately leading to better decision-making and resource utilization.

Simulation Results and Analysis

Simulating a BPMN (Business Process Model and Notation) model for a blockchain reference architecture with swim lanes representing different layers can be a valuable exercise to understand and optimize the interactions and processes within the architecture. In this scenario, the swim lanes represent different layers such as blockchain AI, IoT, prediction layer at the top, blockchain application layer, blockchain layer, and infrastructure layer at the bottom. Additionally, there is a blockchain service layer responsible for coordinating communication among these layers and handling external events. Let's break down how you might approach simulating this architecture using BPMN:

1. Identify Key Processes: Begin by identifying the key processes that occur within each layer and how they interact with each other. For instance, the blockchain AI layer might have processes for data analysis and decision-making, while the IoT layer may involve data collection and transmission.
2. Layer Swim Lanes: Create swim lanes for each layer in your BPMN diagram. Each swim lane represents a separate layer, with the topmost swim lane being the prediction layer, followed by the blockchain application layer, blockchain layer, and infrastructure layer at the bottom. The blockchain service layer can be a separate swim lane or depicted as a coordinating element running through all the layers.
3. Activities and Tasks: Within each swim lane, add activities and tasks that represent the processes and interactions occurring within that layer. For example, in the prediction layer, you might have tasks for data analysis and prediction. In the blockchain application layer, you can include tasks related to smart contract deployment or transaction processing.
4. Gateways and Events: Use gateways and events to represent decision points and triggers in your processes. Gateways can show where conditional flows or splits occur, while events represent occurrences that initiate processes or signal their completion.
5. Message Flows: Use message flows to depict the communication and data exchange between layers. The blockchain service layer is crucial for coordinating communication between different layers and handling external events. Message flows help illustrate the flow of information and requests between these layers.
6. Exclusive and Inclusive Gateways: In a blockchain architecture, there may be situations where you need to make decisions based on certain conditions. Use exclusive and inclusive gateways to model these decision points. Exclusive gateways represent mutually exclusive paths, while inclusive gateways allow multiple paths based on conditions.
7. External Events: Model external events or triggers using BPMN events. These could include events like user interactions, sensor data, or incoming requests from external systems. Show how these events are received and processed within the blockchain service layer and then distributed to the appropriate layers.
8. Data Objects: Represent data objects, data stores, and data flow with the appropriate symbols in BPMN. This helps to visualize how data is collected, stored, and used across the layers.
9. Simulation Tools: Consider using BPMN simulation tools or software that allows you to run simulations of the model. This will help you analyze how the architecture functions under various scenarios and conditions.



Simulation Results

Resources

Reference Architecture for Blockchain (REF-ArcBC))

Scenario information

Name	Scenario 1
Time unit	Minutes
Duration	030,00:00:00

Resource	Utilization	Total fixed cost	Total unit cost	Total cost
Cloud Resources	97.09 %	0	0	0
Knowledge Discovery	76.33 %	0	0	0
Blockchain & AI Scientists	93.20 %	0	0	0
Blockchain Security & Sustaninability	95.92 %	0	0	0
Chatbots & chatGPT	98.06 %	0	0	0
Blockchain Analytics	94.97 %	0	0	0
Blockchain Nodes & Network	97.09 %	0	0	0
	Total	0	0	0

Fig 14 *Simulation results*

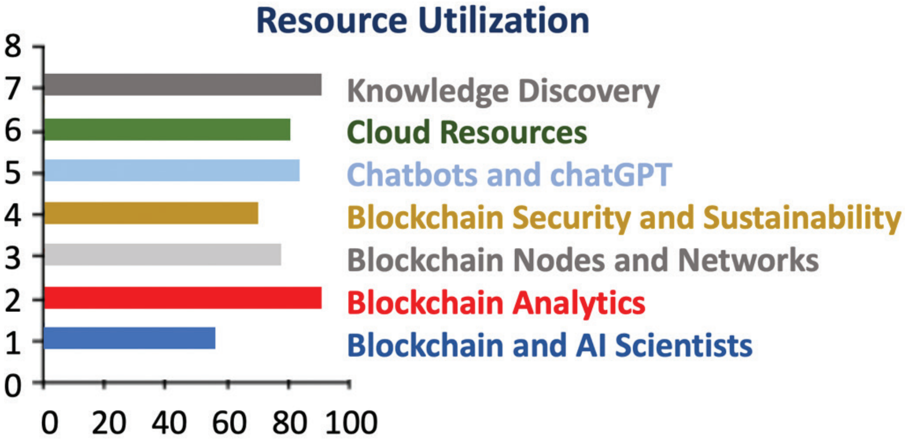


Fig. 15. Resource utilization results.

10. Optimization: Use the simulation results to identify bottlenecks, inefficiencies, or areas for improvement within the architecture. Adjust the model and processes as needed to optimize the flow and performance of the blockchain reference architecture.

By creating a BPMN model with swim lanes, you can visually represent the interactions and processes within your blockchain reference architecture, making it easier to understand, analyze, and optimize. Simulating the model provides insights into how the architecture behaves and performs in real-world scenarios, enabling you to make informed decisions about improvements and adjustments. Figure 13 shows the real-time scenario of the EHR use case with a simulation showing visually the time it takes, the waiting time, and the completion time at each node. The top layer consists of several processes such as receiving real-time data which is fed to the health analytics layer which filters the raw data and does the cleaning. After processing the data, it then passes onto the chatbot service to interpret the data and answers the requested questions, etc.

The simulation shows it has taken 10.45 min to process 100 instances of real-time data and service requests. Figures 14 and 15 show the resources versus utilization for energy efficiency and sustainability of the BPMN. For this scenario of the EHR and 100 user requests, the simulation has consumed 97.09% of the cloud resources, 76.33% for knowledge discovery, it has used 93.20% of blockchain scientists, etc.

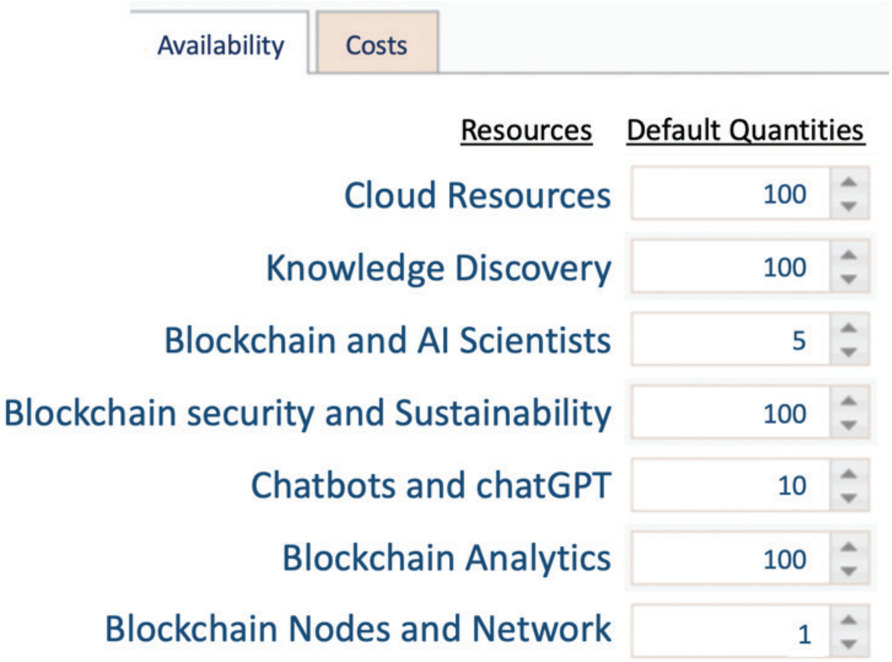
Business Process Model and Notation (BPMN) is a visual modeling language that is often used to represent and analyze business processes within an organization. When it comes to analyzing business processes using BPMN, resource analysis is a critical aspect. Resource analysis allows organizations to evaluate various performance measures to optimize their processes. Here's how BPMN can be used to evaluate the performance measures as follows:

1. Sub- or over-utilization of Resources:
    1. Resource Assignment: In BPMN, you can assign resources to specific tasks or activities using the 'Resource' element. This allows you to identify which resources are used and where they are used.
    2. Resource Pools: Resource pools can be used to represent the availability of resources in the organization. By comparing the assignments to the available resources, you can identify instances of underutilization (resources are not fully utilized) or overutilization (resources are overloaded).
2. Total Resources Costs:
   By associating costs with the resources, BPMN models can help calculate the total resource costs for executing a process. You can assign costs to individual resources and accumulate these costs throughout the process to determine the total resource costs.
3. Total Activity Costs:
   In BPMN, you can also attach costs to activities or tasks. By aggregating these costs across all activities within a process, you can calculate the total activity costs. This includes not only the resource costs but also other costs associated with activities.
4. Delays (Time an Activity Waits for a Resource):
   By modeling the sequence flows between activities in BPMN, you can visually represent the order in which activities are executed. Analyzing these sequence flows allows you to identify delays caused by activities waiting for resources to become available. Resource allocation and availability can be a crucial factor in assessing delays.
5. A More Accurate Expected Cycle Time:
   By considering resource assignments and analyzing the delays, you can calculate a more accurate expected cycle time for a process. This is especially useful for assessing the time it takes to complete a process from start to finish, factoring in resource-related delays. To perform these analyses effectively, BPMN models are often combined with process analysis and simulation tools. These tools can run simulations on the BPMN model to assess resource utilization, costs, and cycle times under different scenarios and resource constraints.

By evaluating these performance measures, organizations can identify areas for improvement in their processes. For example, they can reallocate resources to reduce overutilization or redistribute tasks to prevent underutilization. They can also optimize processes to reduce delays, leading to shorter cycle times and more efficient operations. Additionally, by calculating resource and activity costs accurately, organizations can make informed decisions to optimize their processes and achieve cost savings.

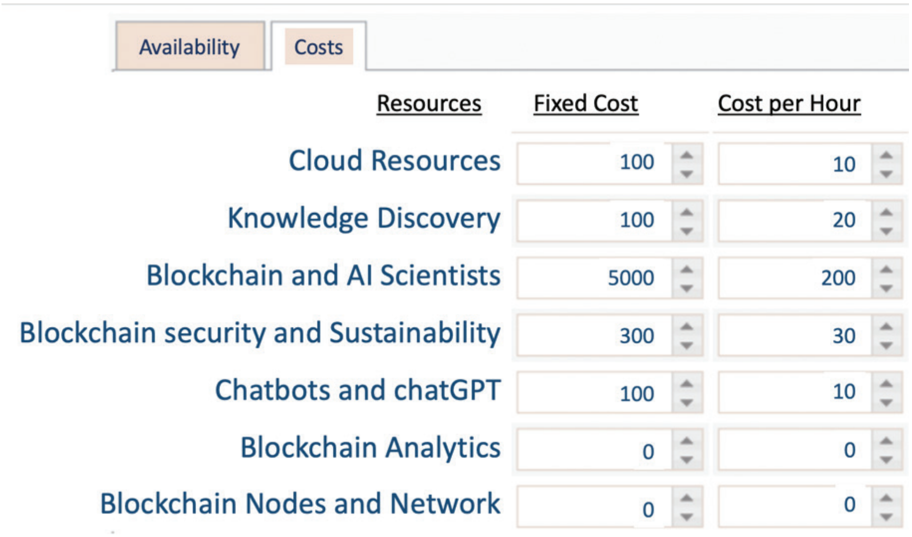
The role of BPMN in conducting simulation experiments is to improve efficiency, reduce costs, and optimize resources in the context of blockchain technology within the healthcare industry. In the healthcare industry today, blockchain technology has emerged as a transformative force. It offers secure and transparent data management, which is critical for the healthcare sector. To ensure its sustainable growth and maximize its benefits, it is essential to continuously assess and optimize various parameters related to its implementation. Figures 16–18 in this context likely refer to graphical representations or visual models created using BPMN. These figures could depict processes, workflows, or other aspects of blockchain technology in healthcare:

1. Parameters for Simulation: The paragraph mentions 'various of these parameters.' These parameters may include resource allocation, process efficiency, cost analysis, and resource utilization. In the healthcare context, these parameters are crucial for ensuring that blockchain technology is effectively integrated and utilized to improve data security, interoperability, and transparency.
2. BPMN as a Simulation Tool: BPMN serves as a powerful tool for modeling and analyzing processes. It allows you to create visual representations of complex workflows, interactions, and resource assignments. These models can be used for conducting simulation experiments. Simulation involves running various scenarios to understand how changes to processes or resource allocation impact efficiency and costs.
3. Simulation Experiments: By conducting simulation experiments using BPMN models, healthcare organizations and stakeholders can test different strategies and scenarios. For instance, they can simulate the impact of reallocating resources within a blockchain-based healthcare system, changing the order of processes, or optimizing data-sharing protocols.
4. Efficiency, Cost, and Resource Optimization: The goal of these simulation experiments is to improve various aspects:



Resources	Default Quantities
Cloud Resources	100
Knowledge Discovery	100
Blockchain and AI Scientists	5
Blockchain security and Sustainability	100
Chatbots and chatGPT	10
Blockchain Analytics	100
Blockchain Nodes and Network	1

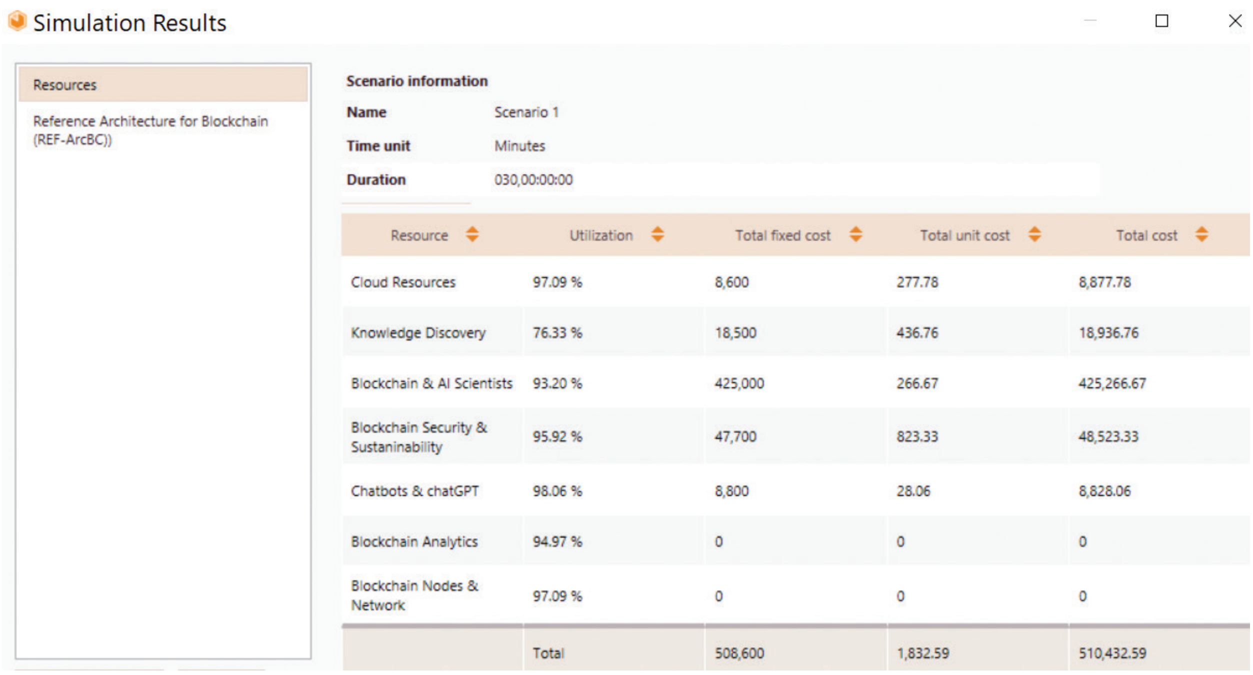
Fig. 16. Resources availability.



Resources	Fixed Cost	Cost per Hour
Cloud Resources	100	10
Knowledge Discovery	100	20
Blockchain and AI Scientists	5000	200
Blockchain security and Sustainability	300	30
Chatbots and chatGPT	100	10
Blockchain Analytics	0	0
Blockchain Nodes and Network	0	0

Fig. 17. Resource utilization cost estimation.

1. Efficiency: By adjusting processes and resource allocation, organizations can identify and implement changes that lead to more efficient operations. This can result in faster data sharing, reduced waiting times, and improved patient care.
2. Cost: Through cost analysis, organizations can identify cost drivers within their blockchain-based healthcare systems. By making data-driven decisions, they can reduce unnecessary expenses and allocate resources more effectively.
3. Resource Optimization: Blockchain technology often involves the allocation of resources such as computing power, storage, and personnel. Simulation experiments allow for the optimization of these resources, ensuring they are used efficiently.
5. Sustainable Growth: Sustainable growth in the context of blockchain technology in healthcare means that the technology is continuously evolving and improving while maintaining its benefits and relevance. By using BPMN for simulation experiments,



Simulation Results

Scenario information

Name	Scenario 1
Time unit	Minutes
Duration	030,00:00:00

Resource	Utilization	Total fixed cost	Total unit cost	Total cost
Cloud Resources	97.09 %	8,600	277.78	8,877.78
Knowledge Discovery	76.33 %	18,500	436.76	18,936.76
Blockchain & AI Scientists	93.20 %	425,000	266.67	425,266.67
Blockchain Security & Sustaninability	95.92 %	47,700	823.33	48,523.33
Chatbots & chatGPT	98.06 %	8,800	28.06	8,828.06
Blockchain Analytics	94.97 %	0	0	0
Blockchain Nodes & Network	97.09 %	0	0	0
	Total	508,600	1,832.59	510,432.59

Fig. 18. Total Cost Estimation.

healthcare organizations can adapt to changing requirements, optimize their systems, and ensure the long-term sustainability and success of their blockchain implementations.

In summary, Figures 16 through 18 likely represent BPMN models that help healthcare organizations conduct simulation experiments to assess and enhance efficiency, reduce costs, and optimize resources in their blockchain technology implementations. This approach enables healthcare to embrace blockchain sustainably, offering long-term benefits in terms of data security, transparency, and interoperability.

Designing Reusable Smart Contracts for EHR

Designing reusable smart contracts for an EHR system is crucial for achieving efficiency, security, and scalability in healthcare data management on a blockchain. Designing reusable smart contracts helps to achieve modularity, scalability, and reusability. The reusable smart contact in HER can then be customized and adopted across all healthcare applications. Designing reusable smart contracts for EHR systems involves a thoughtful combination of blockchain technology, healthcare domain expertise, and a deep understanding of privacy and security considerations. These contracts should not only enhance data management but also promote patient-centric control over their healthcare data while ensuring compliance with healthcare regulations. The Smart Contract for PatientRegistry in Ethereum Solidity is presented in Table 2.

The provided Solidity code defines a smart contract named PatientRegistry that serves as a basic patient information management system on the Ethereum blockchain. Let's break down this code step by step:

1. pragma solidity ^0.8.0;: This line specifies the version of the Solidity compiler that should be used.
2. contract PatientRegistry { ... }: This defines the PatientRegistry smart contract.
3. struct Patient { ... }: This is a data structure that represents a patient's information, including their ID, name, date of birth, gender, contact information, medical history, and allergies. This structure is used to store patient records.
4. uint256 public patientCount: This state variable keeps track of the total number of patients in the registry.
5. mapping (uint256 ≥ Patient) public patients;: This mapping associates patient IDs (of type uint256) with their respective Patient records. It allows you to retrieve patient information by their ID.
6. event PatientAdded (uint256 id, string name, uint256 dateOfBirth, string gender, string contactInformation);: This event is emitted when a new patient is added to the registry. It provides important details of the added patient.
7. function addPatient(string memory _name, uint256 _dateOfBirth, string memory _gender, string memory _contactInformation, string memory _medicalHistory, string memory _allergies) public { ... }: This function is used to add a new patient to the registry. It

Table 2. Smart contract for PatientRegistry in Ethereum solidity.

```
pragma solidity ^0.8.0;
contract PatientRegistry {
 struct Patient {
  uint256 id;
  string name;
  uint256 dateOfBirth;
  string gender;
  string contactInformation;
  vstring medicalHistory;
  string allergies;
 }
 uint256 public patientCount;
 mapping (uint256 => Patient) public patients;
 event PatientAdded (uint256 id, string name, uint256 dateOfBirth, string gender, string contactInformation);
 function addPatient(string memory _name, uint256 _dateOfBirth, string memory _gender, string memory _contactInformation, string memory
 _medicalHistory, string memory _allergies) public {
 patientCount++;
 patients[patientCount] = Patient(patientCount, _name, _dateOfBirth, _gender, _contactInformation, _medicalHistory, _allergies);
emit PatientAdded(patientCount, _name, _dateOfBirth, _gender, _contactInformation);
 }
function getPatient(uint256 _id) public view returns (string memory, uint256, string memory, string memory, string memory, string memory) {
  require(_id > 0 andand _id <= patientCount, “Invalid patient ID”);
  Patient memory patient = patients[_id];
  return (patient.name, patient.dateOfBirth, patient.gender, patient.contactInformation, patient.medicalHistory, patient.allergies);
 }
}
```

takes patient information as arguments and emits the PatientAdded event.

8. patientCount++;: This line increments the patientCount to keep track of the total number of patients.
9. patients[patientCount] = Patient(...);: This line adds a new patient to the patients mapping by associating the patient's ID with their information.
10. function getPatient(uint256 _id) public view returns (string memory, uint256, string memory, string memory, string memory, string memory) { ... }: This function allows you to retrieve a patient's information by providing their ID. It checks if the ID is valid and returns the patient's details as a tuple.
11. require(_id > 0 andand _id <= patientCount, 'Invalid patient ID');: This line ensures that the provided patient ID is within the valid range of IDs stored in the registry. If the ID is out of range, it throws an error.
12. Patient memory patient = patients[_id];: This line retrieves the patient's information from the patients mapping based on the provided ID.
13. return (patient.name, patient.dateOfBirth, patient. gender, patient.contactInformation, patient.medicalHistory, patient.allergies);: This returns the patient's information as a tuple.

In summary, this smart contract, PatientRegistry, allows the addition of patient records and retrieval of patient information based on their unique ID. It's a basic example of how blockchain can be used to manage sensitive healthcare data securely and transparently. Please note that in real-world scenarios, more features, access control mechanisms, and security measures should be considered for the protection of patient data and privacy. In addition, design for reusable smart contracts can boost the productivity and sustainability goals of blockchain. By separating the interface from the actual implementation, you can create other contracts that adhere to the same patient record structure and can use the IRecord interface for interoperability and reusability. For instance, if you want to create another contract that interacts with patient records, you can implement the same IRecord interface and work with patient data seamlessly. This approach promotes code reusability and maintainability.

Conclusion

This paper presents a systematic approach to developing blockchain applications. A systematic framework on S^{3}EF-HBCA has been presented with best practices on requirements engineering for healthcare, business process modeling for healthcare, domain modeling for healthcare, a reference architecture for healthcare, and validation by a case study on EHR Management System, and simulation with BPMN tools. The simulation shows it has taken 10.45 min to process 100 instances of real-time data and service requests. The overall result shows encouragement in terms of process, tools, standards, and testing.

Funding Statement

This research is not funded by any organization or government.

Acknowledgment

The author would like to thank ChatGPT 3.5 for re-writing and proofreading some of the texts in this paper.

Blockchain Concepts Glossary

Application Programming Interface (API): A set of rules and protocols that allows one software application to interact with another, facilitating integration and communication between different systems.

Blockchain for Software Engineering: Adopts the application of blockchain to improve the quality and security of software development processes and assets.

Blockchain: A mathematical structure for storing digital transactions or data in an immutable, distributed, decentralized digital ledger consisting of blocks that are linked via cryptographic signature that is nearly impossible to fake, hack, or disrupt.

Consensus Algorithm: A mechanism used to achieve agreement on a single data value among distributed processes or systems, crucial for validating transactions on a blockchain.

Consensus Layer: The part of a blockchain system responsible for achieving agreement on the state of the blockchain among participating nodes.

Cryptocurrency: A mathematical structure for storing digital transactions or data in an immutable, distributed, decentralized digital ledger consisting of blocks that are linked via cryptographic signature that is nearly impossible to fake, hack or disrupt.

DApp (Decentralized Application): Software applications that run on a decentralized network, typically a blockchain, and use smart contracts for their logic.

Decentralization: The distribution of control and decision-making across a network, reducing reliance on a central authority.

Decentralized, Distributed Ledger: Records transactions across a network of computers, ensuring transparency and immutability.

Digital or virtual currency: Uses cryptography for security and operates on a decentralized network, often based on blockchain technology.

Fork: A split in the blockchain resulting in two separate chains, usually due to a change in the protocol or a disagreement within the community.

Gas: The unit representing the computational effort required to execute operations or transactions on a blockchain network, often associated with transaction fees.

Immutable Code: Code that, once deployed on a blockchain, cannot be changed or updated, emphasizing the importance of thorough testing and security in the development process.

Immutable Ledger: A ledger that cannot be altered or tampered with once a block is added to the blockchain, ensuring data integrity.

Interoperability: The ability of different blockchain networks and software systems to communicate, exchange data, and operate together seamlessly.

Mining: The process of validating transactions and adding them to the blockchain by solving complex mathematical problems, typically associated with proof-of-work consensus algorithms.

Nodes: Devices on a blockchain network that participate in maintaining the distributed ledger by validating and relaying transactions.

Oracles: External agents or services that provide real-world data to a blockchain smart contract, enabling it to make decisions based on information outside the blockchain.

Permissioned Blockchain: A blockchain where access to participate in the network and perform certain actions is restricted to a predefined group of participants.

Scalability: The ability of a blockchain network to handle an increasing number of transactions or users without sacrificing performance.

Self-executing contracts: Terms of the agreement directly written into code, automating and enforcing contractual agreements on a blockchain.

Smart Contract:

Software Development Kit (SDK): A collection of tools, libraries, and documentation that helps developers create software applications for a specific platform or framework.

Software Engineering for Blockchain Glossary: Software Engineering (SE) for Blockchain which adopts systematic processes and techniques for developing blockchain applications.

Tokenization: The process of converting rights to an asset into a digital token on a blockchain, facilitating ownership and transfer.

Wallet: Digital tools that allow users to store and manage their cryptocurrencies, providing private and public key pairs for transactions.

Zero-Knowledge Proof: A cryptographic method that allows one party to prove the authenticity of information without revealing the actual data, enhancing privacy and security in transactions.

References

1. Porru S, Pinna A, Marchesi M, Tonelli R. Blockchain-oriented software engineering: challenges and new directions [Internet]. 2017 [cited 2023 October 10]. Available from https://www.researchgate.net/publication/313844963
2. CompTIA. Blockchain terminology: a glossary for beginners [Internet]. 2023 [cited 2023 September 24]. Available from: https://connect.comptia.org/content/articles/blockchain-terminology
3. Banafa A. Blockchain technology and applications. New York, NY: River Publishers; 2020.
4. Destefanis G, Marchesi M, Ortu M, Tonelli R. Smart contracts vulnerabilities: a call for blockchain software engineering?, 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), Campobasso, Italy, 2018.
5. Beller M, Hejderup J. Blockchain-based software engineering. Technical report. Delft University of Technology; 2018.
6. Chung L, do Prado Leite JCS. On non-functional requirements in software engineering. In: Conceptual modeling: Foundations and applications. Berlin, Heidelberg: Springer; 2009, pp. 363–379.
7. Agbo CC, Mahmoud QH, Eklund JM. Blockchain technology in healthcare: a systematic review. Healthcare, MDPI 2019;7(2):56. https://doi.org/10.3390/healthcare7020056
8. Mayer AH, da Costa CA, Righi RDR. Electronic health records in a blockchain: a systematic review. Health Inform J. 2019;26(1):146045821986635. https://doi.org/10.1177/1460458219866350
9. Christidis M, Devetsikiotis M. Blockchains and smart contracts for the internet of things. 2016. IEEEAccess, Digital Object Identifier.
10. Khezr S, Moniruzzaman Md, Yassine A, Benlamri R. Blockchain technology in healthcare: a comprehensive review and directions for future research. Appl Sci. 2019;9(9):1736. https://doi.org/10.3390/app9091736
11. Hasselgren A, Kralevska K, Gligoroski D, Pedersen SA, Faxvaag A. Blockchain in healthcare and health sciences—a scoping review. Int J Med Inform. 2020;134:104040. https://doi.org/10.1016/j.ijmedinf.2019.104040
12. Tang Y, Xiong J, Becerril Arreola R, Lakshmi L. Blockchain ethics research: a conceptual model. SIGMIS-CPR '19, June 20–22, 2019, Nashville, TN.
13. De Filippi P, Wright A. Blockchains, bitcoin, and decentralized computing platforms. In Blockchain and the law: The rule of code (pp. 13–32). Cambridge, MA: Harvard University Press; 2018.
14. Giungato P, Rana R, Tarabella A, Tricase C. Current trends in sustainability of bitcoins and related blockchain technology. Sustainability. 2017;9(12):2214. https://doi.org/10.3390/su9122214
15. Viriyasitavat W, Hoonsopon D. Blockchain characteristics and consensus in modern business processes. J Indust Inform Integr. 2019;13:32–39. https://doi.org/10.1016/j.jii.2018.07.004
16. Hakak S, Khan WZ, Gilkar GA, Imran M, Guizani N. Securing smart cities through blockchain technology: architecture, requirements, and challenges. IEEE Netw. 2020;34(1):8–14. https://doi.org/10.1109/MNET.001.1900178
17. Vacca A, Di Sorbo A, Visaggio CA, Canfora G. A systematic literature review of blockchain and smart contract development: techniques, tools, and open challenges. J Syst Soft. 2021;174:110891. https://doi.org/10.1016/j.jss.2020.110891
18. Ramachandran M. Software security engineering: design and applications. New York, NY: Nova Science Publishers; 2012.
19. Dzhalila D, Siahaan D, Fauzan R, Asyrofi R, Karimi MI. A systematic review on blockchain technology in software engineering. J ELTIKOM J Teknik Elektro. 2023;7(1):38–49. https://doi.org/10.31961/eltikom.v7i1.725
20. Mean NR, Stehney T. Security quality requirements engineering (SQUARE) methodology. ACM SIGSOFT Soft Eng Notes. 2005;30:1–7. https://doi.org/10.1145/1082983.1083214
21. Feist J, Grieco G, Groce A. Slither: a static analysis framework for smart contracts. IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), 2019. https://doi.org/10.1109/WETSEB.2019.00008
22. Singh I, Lee S-W. RE_BBC: requirements engineering in a blockchain-based cloud– (BBC) system. 2020.
23. Khatter K, Relan D. Non-functional requirements for blockchain enabled medical supply chain. Int J Syst Assur Eng Manag. 2022;13:1219–31. https://doi.org/10.1007/s13198-021-01418-y
24. Pressman R. Software engineering: A practitioner's approach. 8th edn. New York, NY: McGraw Hill.
25. Ramachandran M. Software components: guidelines and applications. New York, NY: Nova Science Publishers; 2008.
26. Lamsweerde AV. Requirements engineering: From system goals to UML models to software specifications. Hauppauge, NY: Wiley; 2009.
27. Shoaib M, Zhang S, Ali HA. Bibliometric study on blockchain-based supply chain: a theme analysis, adopted methodologies, and future research agenda. Environ Sci Pollut Res. 2023;30:14029–49. https://doi.org/10.1007/s11356-022-24844-2
28. Smiraglia R. Domain analysis for knowledge organization ([edition unavailable]) [Internet]. Elsevier Science; 2015 [cited 2023 October 10]. Available from: https://www.perlego.com/book/1831380/domain-analysis-for-knowledge-organization-tools-for-ontology-extraction-pdf
29. Sommerville I. Software engineering. 10th ed. Paramus, NJ: Pearson; 2015.
30. Lardo A, Corsi K, Varma A, Mancini D. Exploring blockchain in the accounting domain: a bibliometric analysis. Account Audit Accountabil J. 2022;35(9):204–233. https://doi.org/10.1108/AAAJ-10-2020-4995
31. Gupta D. Service point estimation model for SOA Based Projects [Internet]. 2013. Available from: http://servicetech
32. Swan M. Blockchain blueprint for a new economy. Sebastopol, CA: O'Reily; 2015.
33. Raval S. Decentralized applications: Harnessing bitcoin's blockchain technology. Sebastopol, CA: O'Reilly; 2016.
34. Siegel D. Understanding the DAO attack [Internet]. 2016. CoinDesk. [cited 2023 October 10]. Available from: http://www.coindesk.com/understanding-dao-hack-journalists/
35. Takagi S. Organizational impact of blockchain through decentralized autonomous organizations. IJEPS. 2017;12:22–41. https://doi.org/10.1007/BF03405767
36. Marchesi M, Marchesi L, Tonelli R. An Agile software engineering method to design blockchain applications, Software Engineering Conference Russia (SECR 2018) [Internet]. Moscow, Russia, October 12–13, 2018 [cited 2023 October 10]. Available from: https://arxiv.org/ftp/arxiv/papers/1809/1809.09596.pdf
37. Ekblaw A, Azaria A, Halamka JD, Lippman A. A case study for blockchain in healthcare: "medrec" prototype for electronic

health records and medical research data [Internet]. 2016 [cited 2023 October 10]. Available from: https://www.media.mit.edu/publications/medrec-whitepaper/

38. HER. The Office of the National Coordinator for Health Information Technology (ONC), What is an Electronic Health Record (EHR)? [Internet]. 2023 [cited 2023 October 11]. [cited 2023 October 10]. Available from: https://www.healthit.gov/faq/what-electronic-health-record-ehr
39. Chang V, Yian Chen Y, Xu QA, Xiong C. Evaluation and comparison of various business process management tools. Int J Bus Inform Syst. 2023;43(3):281–308. https://doi.org/10.1504/IJBIS.2023.132065