International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03744
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

A Technical Review of Smart Contract Architecture and Execution

Rajesh Tanti
Computer Science and Engineering
OP Jindal University, Raigarh

Abstract :
A **smart contract** is a self-executing program designed to automatically enforce the terms of an agreement between two parties, eliminating the need for intermediaries. Stored on blockchain or other distributed ledger technologies (DLTs), smart contracts ensure high security, immutability, and protection from vulnerabilities. These contracts facilitate various transactions, including financial exchanges, service delivery, and data manipulation, such as updating land titles. Additionally, they can be used to enforce privacy protection by selectively releasing privacy-protected data.While smart contracts are not legally binding by default, they automate business processes based on pre-defined conditions. Legal steps must be taken to make them enforceable in a legal context. This paper provides a comprehensive analysis of the anatomy of smart contracts, focusing on their key components, architectural structure, and underlying working principles., aiming to provide a comprehensive understanding of their operation and the challenges they present for adoption in various industries. Additionally, the study examines the various types of smart contracts, their real-world applications, and the significant benefits they offer, including automation, transparency, and security. The paper also addresses the challenges and limitations associated with smart contract implementation, such as scalability issues, security vulnerabilities, and legal complexities. By providing a detailed exploration of smart contracts from design to deployment, this research aims to offer valuable insights into their transformative potential in the digital economy.

Keywords: Blockchain, Smart Contract,dApps,Hyperledger, DeFi,NFT,GDPR.

Introduction:

Nick Szabo, a U.S.-born computer scientist who developed a virtual currency dubbed "Bit Gold" in 1998, a decade before Bitcoin was introduced, was the first to propose smart contracts in 1994. Szabo characterized smart contracts as digital transaction mechanisms that implement a contract's terms.Many predictions made by Szabo in his paper are now a part of our daily lives in ways that precede blockchain technology. However, this idea couldn't be implemented because the necessary technology, primarily the distributed ledger, did not exist then. In 2008, Satoshi Nakamoto introduced the revolutionary blockchain technology in a whitepaper. It prevented transactions from being specified in another block. However, the emergence of cutting-edge technologies acted as stimuli for the rise of smart contracts. Five years on, the Ethereum blockchain platform made practical use of smart contracts achievable. Ethereum is still one of the most prevalent platforms enabling smart contract implementation.The advent of blockchain technology has revolutionized various industries by offering a decentralized, transparent, and immutable ledger system. At the heart of this innovation are smart contracts—self-executing contracts with the terms of the agreement directly written into code. Smart contracts enable automated, secure, and trustless transactions, eliminating the need for intermediaries and reducing the risks of human error. However, while the benefits of smart contracts are undeniable, the growing complexity and the rapid adoption of blockchain applications have raised significant concerns about their security, authentication, and reliability.

This paper aims to dissect the anatomy of smart contracts, exploring their fundamental components, structural design, and the principles that govern their functionality. Furthermore, it delves into the various smart contract platforms, compares their capabilities, and analyzes the types of smart contracts based on their use cases.Through this research, we aim to provide a holistic view of smart contracts, shedding light on their benefits, challenges, and the evolving landscape of blockchain-based applications. By understanding the core aspects of smart contract design and implementation, stakeholders can better harness their potential to drive digital transformation in an increasingly decentralized world.Through a comprehensive analysis, this paper aims to contribute valuable insights to the ongoing discourse on blockchain security, providing developers, researchers, and practitioners with actionable knowledge on building secure and efficient smart contract systems.Fig..1 Showing the Basic structure diagram of Smart contract .

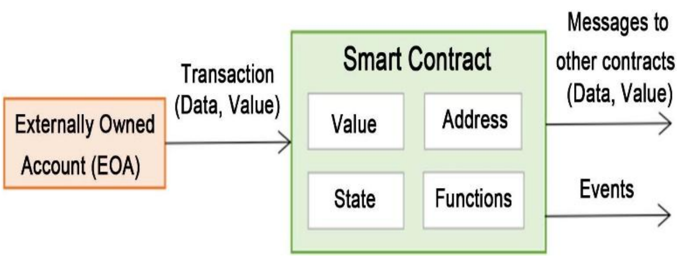

Fig. 1. A basic structure of Smart Contract

literature review /survey

Current reviews on blockchain-enabled smart contracts primarily address concerns such as security, privacy, and performance, yet they do not offer a classification system for enhancing smart contracts or their use in specific domains. Table 1 presents the existing survey of smart contracts. This study builds on earlier surveys by exploring application areas, evaluating challenges, and pinpointing research gaps for future investigations. Table 1 is the Existing smart contract Reviews/Surveys: A comparative summary**.**

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03744
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Key Components of a Smart Contract

A smart contract consists of several key components:

1. Participants: These are the entities that interact with the contract. Participants can be people, systems, or other smart contracts.
2. State: This is the current status of the contract. The state changes as participants interact with the contract.
3. Participants: These are the entities that interact with the contract. Participants can be people, systems, or other smart contracts.
4. State: This is the current status of the contract. The state changes as participants interact with the contract.

Table 1 Existing smart contract Reviews/Surveys: A comparative summary

Survey	Solution taxonomy?	Blockchain platforms?	Application domains?	Coverage of tools?	Research gap identification?	Scope of literature review
Our Servey	√	√	√	√	√	Until 04/2025
ShafaqKhan and Faiza Loukil (6)	√	√	√	√	√	Until 09/2020
Atzei et al. [11]	√	√		√		Until 2017
DHarz and Knottenbelt [13]		√		√		Until 09/2018
Angelo and Salzer [12]		√		√		Until 10/2018
Liu and Liu [33]	√	√		√		2015-2018
Feng et al. [14]		√		√		Until 05/2019
Murray and Anisi [34]		√		√		Until 2019
Zou et al. [18]				√		Until 2019
Gupta et al. [15]	√			√		2015-01/2020
Praitheeshan et al. [20]	√	√		√	√	Until 2019
Mohanta et al. [19]		√	√			Until 2018
Rouhani and Deters [21]		√	√			Until 04/2019
Bartoletti and Pompianu [23]	√	√	√			2013-2016
Meng et al. [37]			√			Until 2018
Hu et al. [25]		√	√			2015-2019
Maesa et al. [38]		√				Until 2018
Cuccuru [24]		√	√			Until 05/2016
Alharby et al. [22]		√	√			Until 05/2017
Udokwu et al. [26]			√			2013-2018
Wang et al. [35]		√	√		√	Until 2018
Zheng et al. [36]		√	√		√	Until 2019

5. Functions: These are the operations that the contract can perform. Functions are triggered by participants and can change the state of the contract.
6. Rules: These are the conditions that govern how the contract operates. Rules are written into the contract's code and must be satisfied for functions to be executed.

Structure of a Smart Contract

A smart contract's structure can vary depending on its purpose, but most smart contracts follow a similar structure:

1. Preamble: This section includes basic information about the contract, such as the contract's name and version.
2. State Variables: These are the variables that store the contract's state. For example, in a smart contract for a sale, there might be state variables for the buyer, seller, price, and item status.
3. Functions: This section contains the functions that the contract can perform. Functions might include actions like initiating the sale, confirming payment, and delivering the item.
4. Modifiers: These are conditions that must be met for functions to be executed. For example, a function might only be executable if the item status is 'for sale'.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03744
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

5. Events: These are actions that trigger updates to the contract's state. Events are logged in the blockchain, providing a transparent record of the contract's activity.

To declare a contract, use the contract keyword followed by the name of the new contract. The body of the contract should be enclosed within curly braces. Each contract is composed of three sections:

Data - specifies the input data, including variable names and types.
Conditions - checks the accuracy of the input data.
Action - details the actions executed by the contract**.**

```
contract MyContract {
   data {
           FromId int
           ToId   int
           Amount money
           }
 func conditions {
                ...
           }
 func action {
            }
          }
```

Fig. 2. A structure of Smart Contract

The above figure 2 showing the structure to create an smart contract .

Architecture of smart contract

The following figure shows the architecture of samart contract , which help us to develop a smart contract and to know about the all procedure of smart contract. **This Architecture** refers to the structured design and framework that defines how smart contracts are developed, deployed, and executed on a blockchain network. It encompasses the components, processes, and interactions required to ensure that smart contracts function securely, efficiently, and as intended without the need for intermediaries. A smart contract, in its modern form as depicted in Figure 2, is a program that is stored and run by a network of computers (known as nodes). These nodes are responsible for tracking the ownership of digital assets, which are linked to accounts. These accounts can belong to either individuals or other smart contracts.

A smart contract operates based on predefined conditions set for incoming transactions. When these conditions are met, the contract automatically triggers specific actions, such as initiating new transactions. The accounts that hold these assets are separate from the nodes; nodes are simply devices that execute the platform's code and do not necessarily own any assets. However, since nodes often receive payments in digital assets for their computational work, they usually maintain their own accounts.

The nodes have key roles: validating and processing transactions, and executing smart contract instructions, which can lead to the creation of new transactions. To ensure agreement on asset ownership, most systems rely on a blockchain—a digital ledger that records every asset transaction from the network's inception.

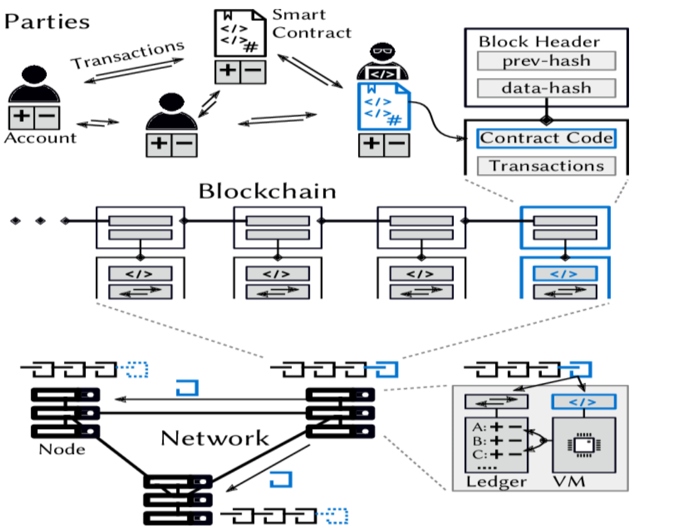

Figure 3: Architecture of a smart contract platform. Contract code and transactions between accounts are stored in a blockchain which is replicated and processed by network nodes.

<u>Working principal of Smart contract :</u>

The **smart contract** is a type of digital agreement that operates through automated computer programs. These programs use mathematical rules to manage and execute transactions, ensuring that the terms of the contract are carried out without the need for a middleman.
Think of it as a set of instructions written in code that automatically performs actions—like transferring money or granting access—when certain conditions are met. This process happens over secure computer networks, often using blockchain technology, which keeps the data safe and tamper-proof.
In simple terms, it's like a digital robot that follows a specific set of rules to complete tasks securely and automatically.
Essentially, smart contracts are computer programs that manage transactions based on pre-set conditions. Szabo suggested that key elements of contracts—like property rights, security deposits, and obligations—should be written into code and run on both hardware and software systems. This reduces the need for trusted third parties, making the process more secure against fraud or hacking.

When smart contracts are built on **blockchain technology**, they become scripts stored on the blockchain. These contracts are self-executing and can be triggered automatically when certain conditions are met. Each smart contract has a unique address on the blockchain, and anyone can send a transaction to interact with it.(refer fig-4)

In below figure 4 we can seen the example Bob who wants to sell her house to John. Instead of traditional paperwork, they use a smart contract to automate the entire process.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03744
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

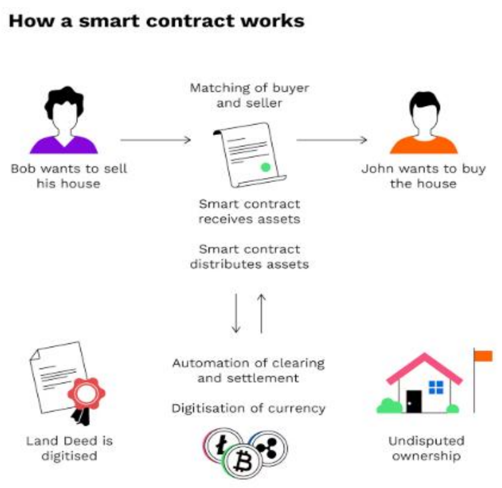

Fig. 4 Working concept of Smart Contract

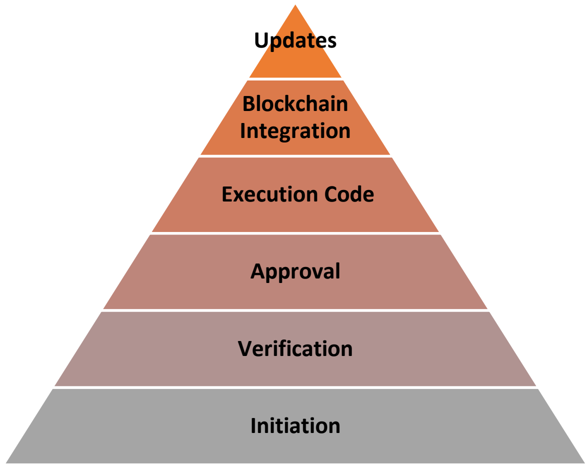

Fig. 5 Pyramid diagram for step by step working procedure of Smart Contract

How Smart Contracts Works

Smart contracts operate on blockchain technology, ensuring transactions are secure, transparent, and automated. Here's a breakdown: (Refer Fig-5)

1. **Initiation:**
   - A user initiates a transaction from their blockchain wallet, such as sending cryptocurrency or triggering a contract.
   - Example: A person wants to buy a car using a smart contract.
2. **Verification:**
   - The transaction is broadcasted to the blockchain network where nodes (computers) verify the sender's identity, balance, and the authenticity of the contract terms.
   - This process uses consensus mechanisms (like Proof of Work or Proof of Stake) to ensure legitimacy.
3. **Approval:**
   - Once verified, the transaction is approved based on the contract's conditions.
   - Example: The buyer's funds are verified to ensure they have enough to complete the purchase.
4. **Execution Code:**
   - The contract contains code (written in languages like Solidity for Ethereum) defining the rules and actions.
   - Example: If the payment is confirmed, the contract automatically transfers ownership of the car to the buyer.
5. **Blockchain Integration:**
   - The transaction, along with the contract code, is added as a new block on the blockchain, creating an immutable record.
   - This ensures the contract cannot be altered without network consensus.
6. **Updates:**
   - Any changes to the contract's status (like updating property ownership) trigger the same verification and execution process, maintaining data integrity.

Popular Smart Contract Platforms:

- **Ethereum:**
  - **Language:** Solidity
  - **Features:** Most popular platform, supports decentralized applications (dApps) and DeFi projects.
  - **Use Case:** Creating ICOs, NFTs, and DeFi protocols.
- **Hyperledger:**
  - **Type:** Permissioned blockchain (not public like Ethereum).
  - **Features:** Ideal for enterprise solutions, focusing on privacy and scalability.
  - **Use Case:** Supply chain management, finance, healthcare data management.
- **Counterparty:**
  - **Type:** Built on Bitcoin's blockchain.
  - **Features:** Adds smart contract capabilities to Bitcoin's infrastructure.
  - **Use Case:** Asset creation, tokenization, and decentralized finance on Bitcoin.
- **Polkadot:**
  - **Type:** Multi-chain network (parachains).
  - **Features:** Connects multiple blockchains for interoperability, enabling scalable transactions.
  - **Use Case:** Cross-chain asset transfers, multi-chain dApps.

Smart Contract Applications :

1. **Records:**
   - **Example:** Healthcare systems use smart contracts to securely store and manage patient data, ensuring only authorized personnel can access it.
   - **Benefit:** Reduces data breaches and ensures real-time updates without human error.
2. **Trade:**
   - **Example:** International trade agreements can be automated, where payments are released automatically once goods are delivered and verified.
   - **Benefit:** Reduces transaction time from weeks to days and lowers costs.
3. **Supply Chains:**

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03744
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

- **Example:** In the food industry, IoT devices track products from farm to table. Smart contracts trigger payments once delivery is confirmed.
- **Benefit:** Improves traceability, reduces fraud, and minimizes food waste.

4. **Mortgages:**
   - **Example:** A smart contract handles mortgage payments automatically. Once conditions are met, ownership is transferred without a notary.
   - **Benefit:** Speeds up the buying process, reduces paperwork, and ensures secure transactions.
5. **Property Market:**
   - **Example:** Smart contracts can register ownership of digital assets like virtual real estate in the metaverse.
   - **Benefit:** Simplifies ownership transfers, reduces legal costs, and enhances transparency.

6. Human Resources (HR):
   - **Use Case:** Smart contracts can securely record and verify academic qualifications, certifications, and work experience.
   - **Benefits:**
     - Prevents CV fraud and misrepresentation.
     - Streamlines the recruitment process with automated background checks.
     - Ensures trustworthy candidate data for employers.
7. **Intellectual Property (IP):**
   - **Use Case:** Smart contracts track ownership and usage rights of patents, copyrights, and trademarks.
   - **Benefits:**
     - Reduces legal disputes over IP ownership.
     - Automates royalty payments and licensing agreements.
     - Enhances transparency in IP management.
8. **Healthcare:**
   - **Use Case:** Smart contracts manage health data, trace medicines, and ensure proper cold chain logistics. They also support health passports and clinical research data integrity.
   - **Benefits:**
     - Enhances data security and patient privacy.
     - Reduces fraud in drug distribution and clinical trials.
     - Automates billing and insurance claims for healthcare services.
9. **Elections:**
   - **Use Case:** Smart contracts ensure secure voter identity verification and tamper-proof vote recording.
   - **Benefits:**
     - Reduces risks of electoral fraud and manipulation.
     - Enhances transparency and trust in election outcomes.
     - Enables remote, secure voting systems with real-time results.
10. **Insurance:**
    - **Use Case:** Automates claims processing, where smart contracts trigger payments based on pre-defined conditions like accident verification or policy compliance.
    - **Benefits:**
      - Speeds up claim approvals, reducing administrative costs.
      - Increases transparency and reduces fraud.
      - Enhances customer satisfaction with faster payouts.

Types of Smart Contracts:

1. **Smart Legal Contracts:**
   - **Definition:** Legal agreements encoded into blockchain, operating on the principle of "If this happens, then that will happen."
   - **Features:**
     - Legally binding with digital signatures.
     - Transparent, immutable, and auditable.
     - Automatically executed when conditions are met (e.g., triggering a payment when a deadline is reached).
     - **Legal Implications:** Failure to comply can result in legal consequences.
   - **Example:** Real estate transactions where payment is released only when property documents are verified.
2. **Decentralized Autonomous Organizations (DAOs):**
   - **Definition:** Organizations governed by smart contracts, enabling democratic decision-making without central leadership.
   - **Features:**
     - Decisions are made through voting mechanisms embedded in the contract.
     - No CEO or board; governance is decentralized.
     - Funds and operations are managed transparently via blockchain rules.
   - **Example: VitaDAO** — A DAO for community-driven scientific research funding.
3. **Application Logic Contracts (ALCs):**
   - **Definition:** Smart contracts that manage interactions between applications, devices, or other contracts, often in IoT ecosystems.
   - **Features:**
     - Operate autonomously between machines rather than people.
     - Used for automating processes in decentralized networks.
     - Sync with other blockchain-based systems for seamless operation.
   - **Example:** Smart contracts managing energy distribution in smart grids or IoT devices triggering actions based on sensor data.

Uses of Smart Contracts

The uses of smart contracts are wide and varied, spread across industries.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
DOI: 10.55041/ISJEM03744
Access || Indexing in all major Database & Metadata

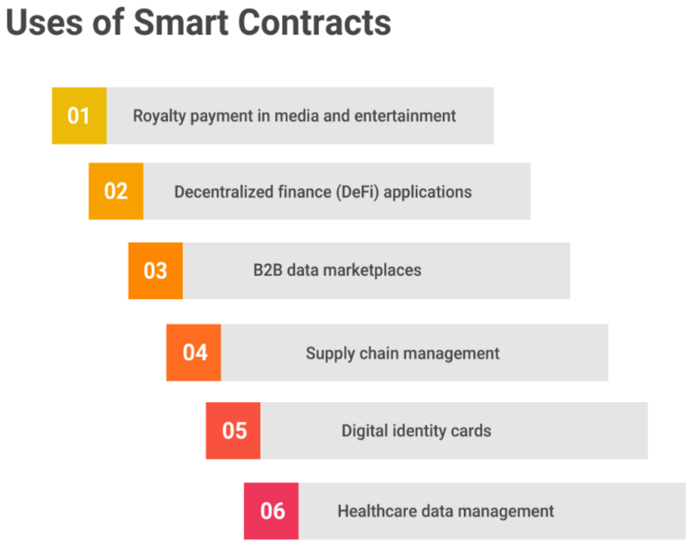

Fig-4. Smart Contracts Uses

1. Royalty payment in media and entertainment

As they enter the industry, new artists rely on revenues from streaming services. Smart contract apps can facilitate easier royalty payments. These contracts can outline, for instance, the share of royalties payable to the record company and the artist. Instantaneous handling of these payments is an enormous advantage for everyone involved.

Smart contracts could also potentially solve the challenge of royalty distribution in an over-the-top (OTT) content world where traditional network agreements do not apply. This technology allows emerging artists and lesser-known actors to get small but regular payments.

2. Decentralized finance (DeFi) applications

Using cryptocurrencies and smart contracts, DeFi apps can offer financial services without an intermediary. DeFi is no longer limited to peer-to-peer transactions. On DeFi platforms, smart contracts facilitate complex processes like borrowing, lending, or derivative transactions.

3. Conversion of assets into non-fungible tokens (NFTs)

By assigning ownership and administering the movable nature of digital assets, smart contracts have made it possible to create non-fungible tokens (NFTs). Contracts like this can also be altered to include added stipulations, like royalties, along with access rights to platforms or software. Essentially, smart contracts make it possible to treat digital assets just like physical ones, with real tangible value.

4. B2B data marketplaces

A data marketplace is a portal where users can buy and sell diverse datasets or data streams from a wide range of sources. Intelligent contracts facilitate the creation of dynamic and fast-evolving markets that support automated and secure transactions without the hassle of human intervention. Datapace is a good example of this particular smart contract use case.

5. Supply chain management

Smart contracts may work autonomously without mediators or third parties because they are self-executing. An organization can create smart contracts for an entire supply chain. This would not require regular management or auditing. Any shipments received beyond the schedule might trigger stipulated escalation measures to guarantee seamless execution.

6. Digital identity cards

Users can store reputational data and digital assets on smart contracts to generate a digital identification card. When smart contracts are linked to multiple online services, other external stakeholders can learn about individuals without divulging their true identities.

For instance, these contracts may include credit scores lenders can use to verify loan applicants without the risk of demographic profiling or discrimination. Similarly, candidates can share resumes without the risk of gender bias in hiring.

7. Electoral polls

Voting could occur within a secure environment created by smart contracts, minimizing the likelihood of voter manipulation. Due to the encryption, every vote is ledger-protected and extremely difficult to decode. Additionally, smart contracts might boost voter turnout. With an online voting system driven by smart contracts, one can avoid making trips to a polling location.

8. Real estate

Smart contracts can accelerate the handover of property ownership. Contracts can be autonomously created and executed. After the buyer's payment to the vendor, for instance, the smart contract may immediately assign control over the asset dependent on the blockchain's payment record.

9. Healthcare data management

Smart contracts can revolutionize healthcare by making data recording more open and efficient. For instance, they might encourage clinical trials by guaranteeing data integrity. Hospitals can maintain accurate patient data records and effectively manage appointments.

10. Civil law

Smart contracts can also flourish in the legal industry. It can be used to create legally binding business and social contracts. In certain regions of North America, governments have authorized smart contracts for digitized agreements. For example, California can issue marital and birth certificates as smart contracts.

Benefits and Challenges of Smart Contracts

Like any technology, smart contracts have both pros and cons. Here are the benefits of smart contracts first:

Benefits of smart contracts

The key reasons to use smart contracts include:

1. Single source of truth

Individuals have the same data at all times, which reduces the likelihood of contract clause exploitation. This enhances trust and safety because contract-related information is accessible throughout the duration of the contract. Additionally, transactions are replicated so that all involved parties have a copy.

2. Reduction in human effort

Smart contracts don't need third-party verification or human oversight. This provides participants autonomy and independence, particularly in the case of DAO. This intrinsic characteristic of smart contracts offers additional benefits, including cost savings and faster processes.

3. Prevention of errors

A fundamental prerequisite for any contract is that every term and condition is recorded in explicit detail. An omission may result in serious issues in the future, including disproportionate penalties and legal complexities. Automated smart contracts avoid form-filling errors. This is one of its greatest advantages.

ISSN: 2583-6129
DOI: 10.55041/ISJEM03744

4. Zero-trust by default

The entire framework of smart contracts is a step beyond conventional mechanisms. This implies that there's no need to rely on the trustworthy conduct of other parties during a transaction. A transaction or exchange does not necessitate faith as a fundamental component, consistent with zero-trust security standards. Since smart contracts operate on a decentralized network, every aspect of the network is more open, fair, and equitable, with no risk of privilege creep.

5. Built-in backup

These contracts capture essential transactional details. Therefore, whenever your data is used in a contract, it is stored indefinitely for future reference. In an instance of data loss, it is simple to retrieve these properties.

Challenges of smart contracts

Smart contracts, as a developing technology, encounter numerous obstacles, including legal challenges, dependence on "off-chain" resources, issues with immutability, scalability, and consensus mechanisms (refer to Fig. 5).

Fig-5 Smart Contract Challenges

Here are the potential downsides of smart contracts and the challenges to be aware of:

1. Rigidity and inconsistent support

Modifying smart contract protocols is nearly impossible, and fixing code errors can be costly and time-consuming. Even if smart contracts conform to the laws of different countries, it might be tough to guarantee that they are adhered to globally.

2. Difficulty in capturing unquantifiable data

For businesses with quantifiable data, such as finance and agriculture, it is relatively simple to put together smart contracts. However, not all industries use quantifiable metrics, like scenarios where creative work has to be evaluated.

3. Conflict with GDPR

The General Data Protection Regulation (GDPR) guarantees *the right to be forgotten by its citizens.* They can request that digital data about them be deleted. Nevertheless, if a digital legal contract binds an individual, it cannot be erased or redacted.

4. Skills shortage

The creation of smart contracts demands expertise in software engineering. Smart contract development is distinct from traditional software development in that it requires coders with organizational expertise and comprehension of non-traditional programming languages such as Solidity. These skills are hard to come by.

- **Learning Curve**: Some languages can be difficult for beginners to grasp.
- **Evolving Standards**: Keeping up with changes and updates in programming languages can be challenging for developers.
- **Interoperability**: Ensuring different languages and systems can work together effectively is often a concern in software development.

5. Scalability Issues

Finally, there is the question of magnitude and scale. Visa can currently process approximately 24,000 transactions per second. According to Worldcoin's 2023 update, Ethereum, the world's biggest blockchain for smart contracts, can only manage 30 transactions per second.

6. **Legal issues**

Legal challenges in smart contracts are significant, particularly due to the conflict between blockchain's immutability and regulations like the GDPR, which grants the "right to be forgotten." Other issues include:

1. **Legal Diversity:** Varying laws across countries complicate compliance.
2. **Complex Legal Representation:** Difficulty in quantifying subjective legal terms for machine execution.
3. **Regulatory Constraints:** Government regulations may undermine blockchain's decentralized nature, shifting reliance to centralized systems.

These challenges highlight the need for balanced approaches to smart contract development and regulation.

Conclusion:-

In conclusion, smart contracts represent a groundbreaking innovation within the realm of blockchain technology, offering a new paradigm for conducting secure, transparent, and automated transactions without the need for intermediaries. This paper has provided an in-depth exploration of the anatomy of smart contracts, covering their key components, architectural structure, and working principles. By examining popular smart contract platforms such as Ethereum, Binance Smart Chain, and Solana, we've highlighted the diverse ecosystems in which these contracts operate and the unique features they offer.

Moreover, the study has detailed the various types of smart contracts and their real-world applications, showcasing their transformative potential across industries like finance, supply chain, healthcare, and more. While smart contracts present numerous benefits, including enhanced efficiency, reduced costs, and increased trust, they also pose challenges related to scalability, security vulnerabilities, and legal complexities.

As the technology continues to evolve, addressing these challenges will be critical to unlocking the full potential of smart contracts. Future research should focus on improving smart contract security, developing more scalable platforms, and exploring regulatory frameworks to support their global adoption. Ultimately, smart contracts are not just a technological advancement—they are a catalyst for the future of decentralized systems and digital economies.

References:

ISSN: 2583-6129
DOI: 10.55041/ISJEM03744

1. Research on the Architecture of Transactional Smart Contracts Based on Blockchains By Zhixiang Liu, Wenlong Feng Yu Zhang and Chengcheng Zhu
2. Smart Contracts in Blockchain Technology: A Critical Review Hamed Taherdoost
3. An Overview of Smart Contract and Use Cases in Blockchain Technology by Bhabendu Kumar Mohanta , Soumyashree S Panda, Debasish Jena
4. Smart Contract: Attacks and Protections By SARWAR SAYEED, HECTOR MARCO-GISBERT, (Senior Member, IEEE), AND TOM CAIRA
5. School of Computing, Engineering and Physical Sciences, University of the West of Scotland, Paisley PA1 2BE, U.K.
6. Blockchain smart contracts: Applications, challenges, and future trends Shafaq Naheed Khan[1] · Faiza Loukil[2] · Chirine Ghedira-
7. Guegan[3] · Elhadj Benkhelifa[4] · Anoud Bani-Hani[1]
8. Smaconat: Smart contracts in natural language E Regnath, S teinhorst 2018 Forum on Specification & Design Languages (FDL), 2018•ieeexplore.ieee.org
9. https://www.rapidinnovation.io/post/smart-contract-developer-roadmap-2024
10. https://www.fortunebusinessinsights.com/smart-contracts-market-108635
11. Atzei N, Bartoletti M, Cimoli T (2017) A survey of attacks on ethereum smart contracts (sok). In: International Conference on Principles of Security and Trust, Springer, pp 164–186
12. Angelo MD, Salzer G (2019) A survey of tools for analyzing ethereum smart contracts. In: IEEE International Conference on Decentralized Applications and Infrastructures, DAPPCON 2019, newark, CA, USA, April 4-9, 2019, IEEE, pp 69–78
13. DHarz D, Knottenbelt W (2018) Towards safer smart contracts: A survey of languages and verification methods. arXiv preprint arXiv:1809.09805
14. Feng X, Wang Q, Zhu X, Wen S (2019) Bug searching in smart contract. arXiv preprint arXiv:1905.00799
15. Gupta R, Tanwar S, Al-Turjman F, Italiya P, Nauman A, Kim SW (2020) Smart contract privacy protection using ai in cyber- physical systems: Tools, techniques and challenges. IEEE Access 8:24746–24772
16. Han S, Xu Z, Zeng Y, Chen L (2019) Fluid: A blockchain based framework for crowdsourcing. In: Proceedings of the 2019 International Conference on Management of Data. Association for Computing Machinery, New York, pp 1921–1924
17. Loukil F, Ghedira-Guegan C, Boukadi K, Benharkat AN (2018) Towards an end-to-end iot data privacy-preserving framework using blockchain technology. In: International Conference on Web Information Systems Engineering. Springer, Berlin, pp 68–78
18. Zou W, Lo D, Kochhar PS, Le XBD, Xia X, Feng Y, Chen Z, Xu B (2019) Smart contract development: Challenges and opportunities. IEEE Transactions on Software Engineering
19. Mohanta BK, Panda SS, Jena D (2018) An overview of smart contract and use cases in blockchain technology. In: 9Th international conference on computing, communication and networking technologies, ICCCNT 2018, bengaluru, india, july 10-12, 2018, IEEE, pp 1–4
20. Praitheeshan P, Pan L, Yu J, Liu J, Doss R (2019) Security analysis methods on ethereum smart contract vulnerabilities: A survey. arXiv preprint arXiv:1908.08605
21. Rouhani S, Deters R (2019) Security, performance, and appli- cations of smart contracts: A systematic survey. IEEE Access 7:50759–50779
22. Alharby M, Aldweesh A, van Moorsel A (2018) Blockchain-based smart contracts: A systematic mapping study of academic research (2018). In: 2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB), IEEE, pp 1–6
23. Bartoletti M, Pompianu L (2017) An empirical analysis of smart contracts: Platforms, applications, and design patterns. In: International Conference on Financial Cryptography and Data Security, Springer, pp 494–509
24. Cuccuru P (2017) Beyond bitcoin: An early overview on smart contracts. I J Law Inform Technol 25(3):179–195
25. Hu Y, Liyanage M, Mansoor A, Thilakarathna K, Jourjon G, Seneviratne A (2018) Blockchain-based smart contracts- applications and challenges. arXiv preprint arXiv:1810.04699
26. Udokwu C, Kormiltsyn A, Thangalimodzi K, Norta A (2018) The state of the art for blockchain-enabled smart-contract applications.in the organization. In: 2018 Ivannikov ispras open conference (ISPRAS), IEEE, pp 137–144
27. Wang S, Yuan Y, Wang X, Li J, Qin R, Wang F (2018) An overview of smart contract: Architecture, applications, and future trends. In: 2018 IEEE Intelligent vehicles symposium, IV 2018, changshu, suzhou, china, june 26-30, 2018, IEEE, pp 108–113
28. Zheng Z, Xie S, Dai HN, Chen W, Chen X, Weng J, Imran M (2020) An overview on smart contracts: Challenges, advances and platforms. Futur Gener Comput Syst 105:475–491
29. GDPR (2016) Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46. Official Journal of the European Union (OJ) 59:1–88
30. Levi SD, Lipton AB (2018) An introduction to smart contracts and their potential and inherent limitations. In: Harvard law school forum on corporate governance & financial regulation
31. Regnath E, Steinhorst S (2018) Smaconat: Smart contracts in natural language. In: 2018 Forum on specification & design languages (FDL), IEEE, pp 5–16
32. Liu J, Liu Z (2019) A survey on security verification of blockchain smart contracts. IEEE Access 7:77894–77904
33. Murray Y, Anisi DA (2019) Survey of formal verification methods for smart contracts on blockchain. In: 10Th IFIP international conference on new technologies, mobility and security, NTMS 2019, canary islands, spain, june 24-26, 2019, IEEE, pp 1–6
34. Wang S, Yuan Y, Wang X, Li J, Qin R, Wang F (2018) An overview of smart contract: Architecture, applications, and future trends. In: 2018 IEEE Intelligent vehicles symposium, IV 2018, changshu, suzhou, china, june 26-30, 2018, IEEE, pp 108–113
35. Zheng Z, Xie S, Dai HN, Chen W, Chen X, Weng J, Imran M (2020) An overview on smart contracts: Challenges, advances and platforms. Futur Gener Comput Syst 105:475–491
36. Meng W, Wang J, Wang X, Liu J, Yu Z, Li J, Zhao Y, Chow SS (2018) Position paper on blockchain technology: Smart contract and applications. In: International Conference on Network and System Security, Springer, pp 474–483
37. Maesa DDF, Mori P, Ricci L (2019) A blockchain based approach for the definition of auditable access control systems. Computers & Security 84:93–119.