COMMUNICATIONS IN SCIENCE AND TECHNOLOGY
Homepage: cst.kipmi.or.id

AniraBlock: A leap towards dynamic smart contracts in agriculture using blockchain based key-value format framework

Irwansyah Saputra[a,b,*], Yandra Arkeman[c], Indra Jaya[d], Irman Hermadi[a], Nur Arifin Akbar[e], Indrajani Sutedja[f]

[a]*Departement of Computer Science, Institut Pertanian Bogor, Bogor Regency 16680, Indonesia*
[b]*Department of Information System, Nusa Mandiri University, Jakarta 13620, Indonesia*
[c]*Departement of Agro-Industrial Technology, Institut Pertanian Bogor, Bogor Regency 16680, Indonesia*
[d]*Department of Marine Science and Technology, Institut Pertanian Bogor, Bogor Regency 16680, Indonesia*
[e]*Dipartimento Matematica e Informatica, Universitá degli Studi di Palermo, Palermo 90133, Italy*
[f]*Information Systems Department, School of Information Systems Bina Nusantara University, Jakarta 11480, Indonesia*

Article history:
Received: 29 July 2023 / Received in revised form: 30 Oktober 2023 / Accepted: 3 December 2023

Abstract

Blockchain technology offers data transparency and traceability, which is particularly useful in the agricultural sector, especially within the supply chains of commodities like coffee and fish. This sector often encounters issues such as quality degradation, unclear information, and socioeconomic injustice affecting stakeholders. The implementation of Static Smart Contracts (SSCs) on blockchains provides a structured method for executing agreements. However, this approach also has limitations, including a lack of flexibility and responsiveness to dynamic changes in the supply chain. Despite these challenges, blockchain remains a valuable tool for ensuring transaction transparency, traceability, and integrity, which are vital in agriculture. These limitations involve unchangeable parameters, rigid rules, and constraints on adaptability and scalability. This study aims to tackle these issues by designing a more dynamic and responsive smart contract system. We introduce AniraBlock, a revolutionary concept for the agricultural supply chain, particularly in the coffee and fish sectors, by implementing Dynamic Smart Contracts (DSCs) based on a key-value format framework. Unlike SSCs, DSCs offer enhanced adaptability and scalability, addressing the former's limitations. Our study adopts a mixed-method approach, utilizing both qualitative and quantitative data to validate AniraBlock's effectiveness. Preliminary results show significant improvements in data management and supply chain transparency. The proposed framework has the potential to influence the agricultural sector by boosting data integrity and operational efficiency.

Keywords: Blockchain; smart contract, fish; coffee; supply chain

1. Introduction

The introduction of blockchain technology is attributed to Nakamoto [1]. This technology is characterized by its decentralized nature, enhancing the security and transparency of digital transactions [2,3]. This advancement fosters egalitarianism among entities within the system and facilitates secure interactions among network members without requiring a trusted authority [4]. Blockchain's decentralization aids in the transparency and traceability of data for organizations [5]. While commonly associated with Bitcoin, blockchain's utility spans various sectors including finance, transportation, education, logistics, health, insurance, retail, and agriculture.

The agricultural industry employs blockchain technology to improve the accuracy of monitoring agricultural products [5]. The utilisation of blockchain technology in the agricultural and food sectors is becoming increasingly widespread, particularly in rising nations such as India, China, and Indonesia, where notable progress in its acceptance has been witnessed [6]. The focal points of concentration revolved around the fishing and coffee sectors. Fish and its processed derivatives are widely recognised as important contributors to dietary protein intake and are consumed extensively throughout [7–13]. Indonesia, being the second-largest producer of fishery products worldwide, faces various issues, such as the deterioration of fish quality during transit and storage, along with the need for improved transparency on the origin and condition of fish [14–16]. Another concern that emerges is the need for fishermen to receive appropriate compensation for their work [15].

In contrast, coffee is a highly consumed and traded agricultural commodity on a global scale [17]. Nevertheless, coffee supply chains frequently encounter illicit trade activities, instances of price manipulation, and inequitable treatment by growers [17–24]. Moreover, it should be noted that coffee has a limited shelf life and incurs substantial expenses throughout the processing stage [25]. The utilisation of blockchain technology has the potential to address these challenges by establishing a system that is transparent and capable of automatic verification.

* Corresponding author.
Email: 92irwansyah@apps.ipb.ac.id
https://doi.org/10.21924/cst.8.2.2023.1240

This system enables the tracing of product origins and guarantees that the remuneration received by farmers or fishermen aligns with the genuine value of their products [26]. This study examines the difficulties of transparency and efficiency in the fishing and coffee industries. These sectors have considerable economic and social significance in many countries, including Indonesia.

While the implementation of blockchain technology in the agriculture sector, particularly in the fish and coffee industries, offers potential solutions to several issues, it is not without hurdles. The SCS has notable constraints. First, SCS has inherent characteristics, wherein its parameters are fixed and unalterable. These circumstances may lead to constraints in adapting to fluctuations in market conditions or user requirements [27]. Furthermore, regulations within the SCS framework have been established and are immutable. This implies that in the event of legislative or regulatory modifications, SCS may be required to promptly adapt [28]. Furthermore, it is imperative to enhance flexibility in the SCS. The agricultural sector requires a high degree of flexibility because of the frequent and unpredictable changes in conditions [29]. Furthermore, it is not feasible to modify or adjust the SCS. The ability to adjust and respond to shifts in dynamic and fluctuating environments is of paramount importance [29]. Ultimately, the scalability of SCS has certain limitations. In a broader scope, this constraint can pose a significant barrier to the successful execution and effectiveness of operations [30]. This visualisation is shown in Figure 1.

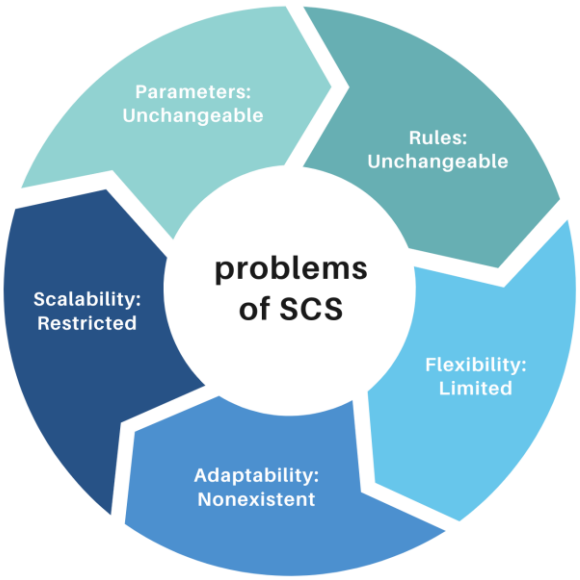

Fig. 1. SCS issues

After delineating the manifold constraints inherent in SCS, as shown in Figure 1, we propose an alternative representation in the form of a non-deterministic finite automaton (NFA), designated in Fig.2, for this study [31]. In this NFA model, each state (e.g. q0, q1, q2) symbolises a functional element or phase within the static contract. For instance, state q0 could correspond to the 'Registration / Role Addition' phase in an SCS, while q2 might pertain to the 'Product Creation' phase [32]. The transition function (δ) was configured by the immutable operations embedded in the SCS, thereby underlining the rigidity of the system. For example, proceeding from the state q0 enables a transition to state q1 via the 'Register / Add Role' operation, denoted by input symbols 0 or 1. Alterations to different operations in this state would necessitate contract modifications. This represents an operational bottleneck, especially in sectors such as agriculture where variables such as commodity pricing or regulations are subject to rapid fluctuations. Hence, the NFA model shown in Figure 2 serves as a visual instrument to accentuate the limitations of SCS, particularly in dynamic and ever-changing contexts. This NFA model serves not only as a visualisation tool but also as a conceptual foundation for the development of SCD, which will be further explored in this study.

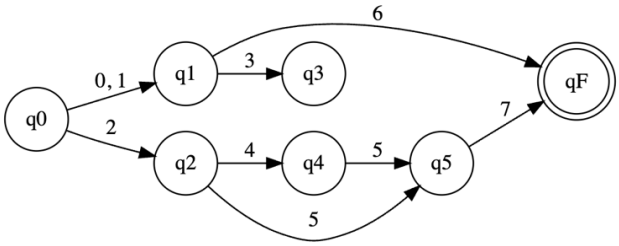

Fig.2. Non-deterministic finite automaton of SCS

Given these constraints, there is a need for a more dynamic and flexible approach that can adapt quickly to changing market conditions and regulations. Therefore, integrating various commodities into a single blockchain platform using SCD based on the key-value format could be a solution. The utilisation of the key-value format is prevalent in computer data storage systems, wherein each value is linked to a distinct and exclusive key [33]. Numerous technologies and applications have used this principle [34]. The objective of this strategy is to enhance the efficacy, dependability, and transparency of the management of coffee and fish commodities. Moreover, SCD can manipulate data more flexibly and cost-effectively. Hence, this study aims to assess the potential of a blockchain-based key-value format to facilitate SCD for the management of coffee and fish commodities in the agricultural industry. The purpose of this study is to propose a more efficient solution for commodity management in this sector.

This study focuses primarily on Indonesia, a country that presents unique challenges and significant opportunities for the incorporation of blockchain technology into its agricultural sector, particularly in the fisheries and coffee industries.

2. Materials and Methods

This study uses several tools, such as Solidity language, which supports the creation of SCS and SCD based on the key-value format [11]. Polygons are then used as a blockchain platform to support smart contracts and provide high transaction speeds at low costs [35]. In addition, Remix software was used to run and test the smart contract during development before it was implemented on Polygon [36]. Finally, Python was used to generate the "parent" smart contract as a basis for the proposed SCD. Python enables the creation and modification of SCD as required [37].

Apart from the materials, the research methodology is designed to respond to the challenges faced in applying smart contracts in the agricultural sector, specifically in the case of coffee and fish commodities. The focus is on creating a blockchain-based solution in the form of an SCD that can improve efficiency and transparency in managing these two commodities. To achieve this goal, this study followed a design process that included five essential stages: Problem Identification, Research (investigation), Solution Design, Prototyping (simulation), and pre-evaluation [38].

This study focuses on the problems that arise from using SCS in coffee and fish agriculture and how to develop an SCD-based solution to overcome these problems. The flow of the methodology used is illustrated in Figure 3.

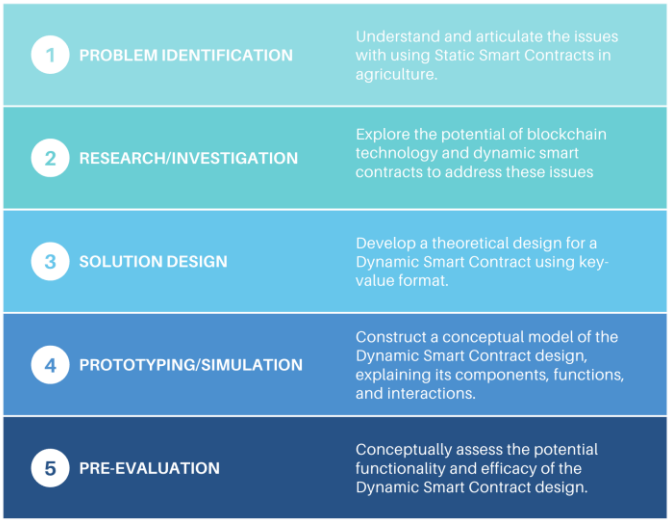

Fig. 3. Research Method

The first stage is Problem Identification, which focuses on recognising, understanding, and elaborating on various challenges and obstacles arising from using SCS, especially in managing coffee and fish commodities. In this phase, every aspect affecting the efficiency and effectiveness of SCS was carefully reviewed and documented. For example, restrictions on flexibility, adaptability, rigid parameters, fixed rules, and limited scalability can be considered.

Next, during the Research/Investigation stage, a deep analysis of blockchain technology's potential and how SCD can overcome the previously identified challenges will be conducted. This phase involved literature research and theoretical studies to understand the concept and operating mechanism of SCD, and how the key-value format can be applied in this context.

Subsequently, in the Solution Design/Creation stage, an initial design sketch or theoretical outline of a key-value format-based SCD was developed. This stage aims to formulate a design capable of overcoming the barriers and challenges identified in the first stage and bringing about significant improvements compared with the existing system. In this stage, each component of the SCD and its interactions are designed and detailed. The fourth stage, Conceptual Prototyping, focuses on creating a conceptual model of the designed SCD. At this stage, a conceptual working model of the designed solution is created, explaining its main components and functionalities and how they interact within the larger system.

In the final stage, a pre-evaluation and theoretical assessment of the potential functionality and efficacy of the SCD design are conducted. At this stage, an analysis is conducted on the effectiveness and efficiency of SCD to improve transparency and efficiency and enhance adaptability in commodity management. This analysis assesses five main aspects: (1) parameters that can change, (2) rules that are more dynamic, (3) increased flexibility, (4) improved adaptability, and (5) wider scalability. Testing of these five aspects will be discussed in future research. Therefore, the Pre-Evaluation stage is crucial for setting focus and testing objectives for subsequent research.

3. Result and Discussion

The previous section delineated several issues inherent to the SCS. One of the primary considerations is the level of rigidity exhibited by the contract parameters. In the context of SCS, the parameters are established and remain unalterable upon the inception of the contract. The lack of flexibility inherent in the system constrains its capacity to effectively respond to dynamic situations frequently observed in many industries, including agriculture. These variables may encompass variations in commodity prices or alterations in government policies among other factors.

The second facet pertains to regulations. Like the parameters, the rules within the SCS framework were predetermined and immutable. This constraint restricts the capacity of the system to adjust to novel or exceptional circumstances that may emerge over time.

The third element under consideration is flexibility. The absence of adaptability exhibited by the SCS has emerged as a significant issue. Owing to its inherent incapacity to alter rules or parameters, SCS exhibits little adaptability in response to evolving requirements or unanticipated circumstances.

The fourth dimension pertained to the concept of adaptation. In the context of SCS, adapting to novel locations or situations is exceedingly challenging. As a result, the diminished efficacy of smart contracts in task execution and the emergence of operational challenges were observed.

Scalability represents the fifth and ultimate dimension. The scalability of SCS is hindered by its design, which mostly caters to operations with well-defined scopes and bounds. The capacity for easy expansion or adjustment to accommodate the more intricate requirements is limited.

Once the problem identification stage is concluded, the subsequent phase involves conducting a study or investigation. During this phase, our objective was to identify potential resolutions to the difficulties encountered by the SCS. After intensive research and investigation, the key-value format could be a potential solution. The key-value format is used in computer data storage, where each value is associated with a unique key [33]. This concept has been widely used in various technologies and applications [34] and the authors argue that it can also be applied to SCD. The structure of the key-value format-based data in the database is shown in Figure 4 [39].

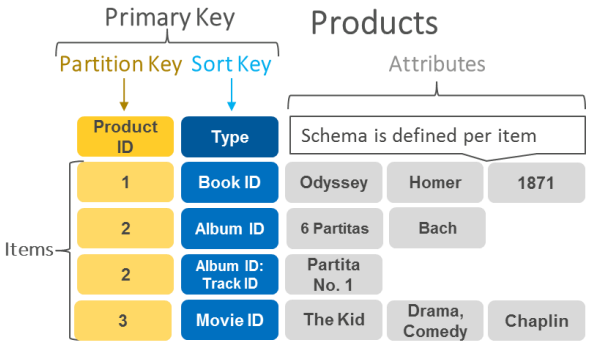

Fig. 4. Key-Value Format Data Structure

For example, a key can represent a particular variable or parameter in SCD. Simultaneously, the value refers to the status or condition of the variable. Using this approach, contracts can be made more flexible and dynamic. The value of a variable can be changed or updated at any time based on the dynamics of the relevant conditions or situations. Thus, SCD offers higher adaptability and flexibility than SCS.

The mapping and storage processes in the SCS occur within a framework set from the beginning. The key-value format in the SCS is predetermined; therefore, it can only accept and store data according to the established structure. For example, if a contract is made for a fish supply chain, the key-value pairs that can be used are only about attributes and transactions related to fish. The

data storage structure also follows a fixed pattern in line with the limitations of the key-value pair format.

Unlike SCS, SCD offers greater flexibility in mapping and storing data. In SCD, the key-value pair format can be changed and adapted based on the user input. For example, if a user enters data about fish, the system generates key-value pairs relevant to fish. If the user enters data on coffee, the system can adapt and create new key-value pairs relevant to coffee. Regarding data storage, SCD can adjust its structure based on the formed key-value pairs, making it capable of accommodating various data types with a dynamic structure. Therefore, SCD offers a more flexible solution and can handle various types of data by adapting to environmental changes and supply chain needs. In addition, the key-value format concept can be used to create a "parent" smart contract that can generate "child" smart contracts to produce various smart contracts from each commodity created, such as fish and coffee.

Having recognised the potential of the key-value format to enhance smart contracts, it is important to undertake a multidimensional analysis of the proposed SCD for a well-rounded understanding. This discussion was structured across six distinct levels. At Level 0, we laid the mathematical foundation for both Static and SCD, elucidating the underlying mathematical principles. Level 1 is dedicated to conceptual modelling through Non-deterministic Finite Automata (NFA), offering a macro-level view of how state transitions occur in SCDs. Level 2 goes under the hood to examine the SCS and SCD data structures, providing insights into information storage and retrieval mechanisms. At Level 3, our exploration extends to the broader agricultural ecosystem, examining how the SCD performs, particularly in the context of coffee and fish commodities. Level 4 is designed to focus on virtual modelling, to illuminate the inner workings and interactions within the SCD framework. This multilevel approach provides an in-depth understanding of the limitations of SCS and demonstrates how key-value-format-based SCD offers a more flexible and robust solution.

In the foundational layer, or Level 0, we discuss mathematical models that provide a quantitative understanding of both SCS and SCD. For SCS, the equation $SCS = \alpha \times \beta \times \gamma \times \delta - \epsilon$ serves as the baseline model, where α stands for the fixed number of nodes or actors, β represents the fixed number of transactions, γ is the fixed number of products or commodities, δ denotes the fixed number of possible actions, and ϵ is the computational cost, often referred to as the Gas Fee. For example, with values $\alpha = 10$, $\beta = 20$, $\gamma = 5$, $\delta = 3$, $\epsilon = 2$, we find that $SCS = 2998$, illustrating that the efficacy of the SCS is 2998 when accounting for the number of actors, transactions, products, actions, and subtracting the computational cost.

For SCD, we extend the model to $SCD = (\alpha + \Delta\alpha) \times (\beta + \Delta\beta) \times (\gamma + \Delta\gamma) \times (\delta + \Delta\delta) - \epsilon \times \zeta$. In this model, $\Delta\alpha, \Delta\beta, \Delta\gamma, \Delta\delta$ represent the flexibility in altering the number of nodes, transactions, products, and actions, while ζ accounts for scalability in terms of transactions per second. To illustrate, let's consider $\alpha = 10, \Delta\alpha = 2, \beta = 20, \Delta\beta = 4, \gamma = 5, \Delta\gamma = 1, \delta = 3, \Delta\delta = 1, \epsilon = 2, \zeta = 1$. Here, the SCD efficacy is calculated as 6910, emphasising that SCDs not only account for the basic parameters but also provide flexibility in terms of scalability and adaptability.

In summary, the mathematical formulations presented in Level 0 offer compelling evidence for the superior efficacy of SCD over its static counterparts. The flexibility and scalability factors of the SCD model significantly elevate its efficiency score, making it a more adaptable and robust solution for diverse applications.

Progressing to Level 1 of our hierarchical analysis, we extend our focus to the use of Non-deterministic Finite Automata (NFA) for SCD. In the NFA for SCD, the set of states Q is defined as {q0, q1, q2, q3, q4, q5, q6, qF}. The letter Σ includes the symbols {0, 1, 2, 3, 4, 5, 6, 7, 8}, each corresponding to different contract actions such as 'Register / Add Role,' 'Create Product', and so on. The transition function δ defines the state changes based on the given alphabetic symbols. The NFA starts at the initial state $q0$ and aims to reach the acceptance state $F=\{qF\}$.

A critical distinction between the NFA models for Static and SCD conditions arises in the alphabet set Σ. Specifically, the alphabet for SCD is extended to include the symbol '9', representing the 'Add/Remove Actor' action, which is absent in SCS. This added functionality infuses another layer of flexibility, exemplifying the adaptability and extensibility that are unique to SCDs.

The transition table also manifests the inherent non-determinism in the SCD. For instance, from state $q1$, a transition can lead to either state $'qF'$ or $q1$ when the input is '6', enabling multiple potential outcomes from the same starting condition. This non-deterministic behaviour sets the SCD apart from its static counterpart by allowing more complex and dynamic interactions. The details of the NFA model are shown in Figure 5.

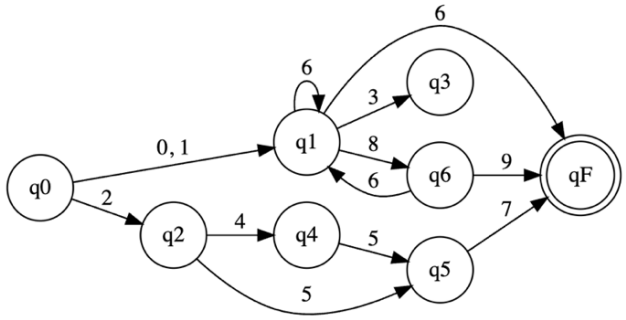

Fig. 5. Non-deterministic Finite Automaton of SCD

Compared to the NFA model for SCS, the SCD NFA features a broader alphabet set Σ and a more complex transition function δ. These intricacies offer greater flexibility and adaptability and are inherently lacking in static models. For example, the addition of the symbol '9' for 'Add/Remove Actor' in the alphabet set Σ for SCDs demonstrates this contract type's ability to dynamically manage participants, a feature not available in SCS. Moreover, the presence of non-deterministic transitions in the NFA of the SCD underlines the capability of the dynamic model to handle a range of outcomes from the same starting state, thereby accommodating more nuanced real-world scenarios.

In conclusion, the NFA representation at Level 1 accentuates the superior flexibility and adaptability of SCD over its static counterparts. These qualities render SCDs a more robust solution for a wider range of applications.

Expanding upon the intricacies of Level 2, our investigation shifts focus to examining the nuanced architectural differences between SCS and SCD. As shown in Figure 5, SCS employs a relatively straightforward data structure. Comprised of basic types, such as **id: string, name: string, quantity: int, and price: decimal**, it represents a rather static model that initiates at the Smart Contract Development stage. This framework is built upon a fixed set of functions, such as **register(), login(), createProduct(), and getproduct()**, which are programmed to interact with the aforementioned data types.

The development process undergoes post-compilation scrutiny wherein a decision node assesses the need for any

modifications to the product data structure. If changes are deemed necessary, the workflow returns to the developmental stage for revisions. In contrast, if the data structure remains stable, the flow advances towards the Application Binary Interface (ABI). Here, ABI serves as a critical juncture, channelling binaries of functions into the stack memory and dispatching binaries of data to both stack and heap memory within the blockchain's runtime environment. Notably, a designated data pathway facilitates the transfer between the stack and heap memory, which can be particularly useful for managing complex data structures or transitioning between different states of data.

To confirm these observations, it is evident that the architectural design of the SCS, as represented in Figure 6, is inherently more rigid and constrained, especially in terms of its data structures and the flow of binaries. This design limits its flexibility in adapting to changes, making it less suitable for applications that require dynamic data management and real-time adaptability.

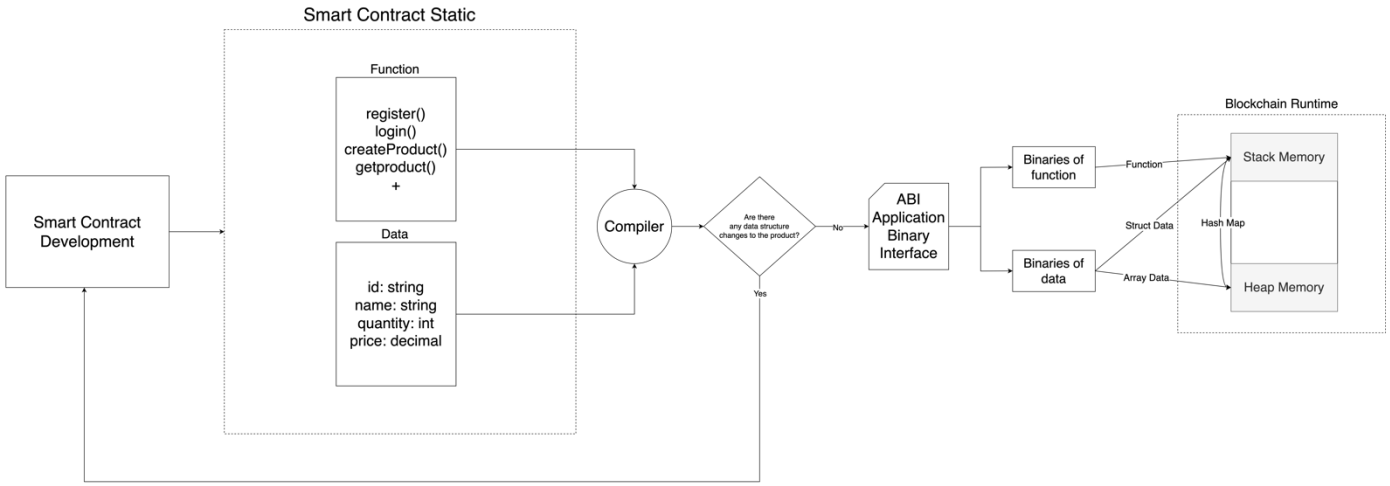

Fig. 6. Architecture Diagram of SCS's Data Structure

Delving deeper into the architecture of SCD, which is illustrated in Figure 7, diverges significantly from the SCS structure in terms of data management. Specifically, SCD leverages Hashmap-String JSON to introduce a more dynamic form of data storage and management. This alternative data structure allows for a broader range of data types and more complex relationships, thereby granting developers a greater degree of flexibility.

Moreover, SCD eliminates the decision node that interrogates the need for any structural changes post-compilation in the SCS. By forgoing this step, SCD facilitates a more streamlined workflow, enabling an uninterrupted transition from the compilation stage to the Application Binary Interface (ABI). This removal not only simplifies the development process but also eliminates the possibility of getting caught in iterative developmental cycles that can delay project timelines.

When it comes to data allocation within the blockchain runtime environment, SCD takes a divergent approach. Unlike SCS, which allocates binaries of data to both stack and heap memory, SCD directs these binaries exclusively to heap memory. This choice eliminates the need for any data transfer pathways between the stack and heap memory, thereby simplifying data management and potentially enhancing performance.

To illustrate this with a practical example, let us consider a case involving fishery management. Within the somewhat restrictive SCS framework, every fish product is assigned distinct variables such as id, name, quantity, and price. If any attributes, such as size or species, need to be updated or added, this would necessitate a reiteration of the development cycle. On the other hand, SCD's Hashmap String JSON accommodates dynamic attribute modifications effortlessly without requiring a return to the initial development stages.

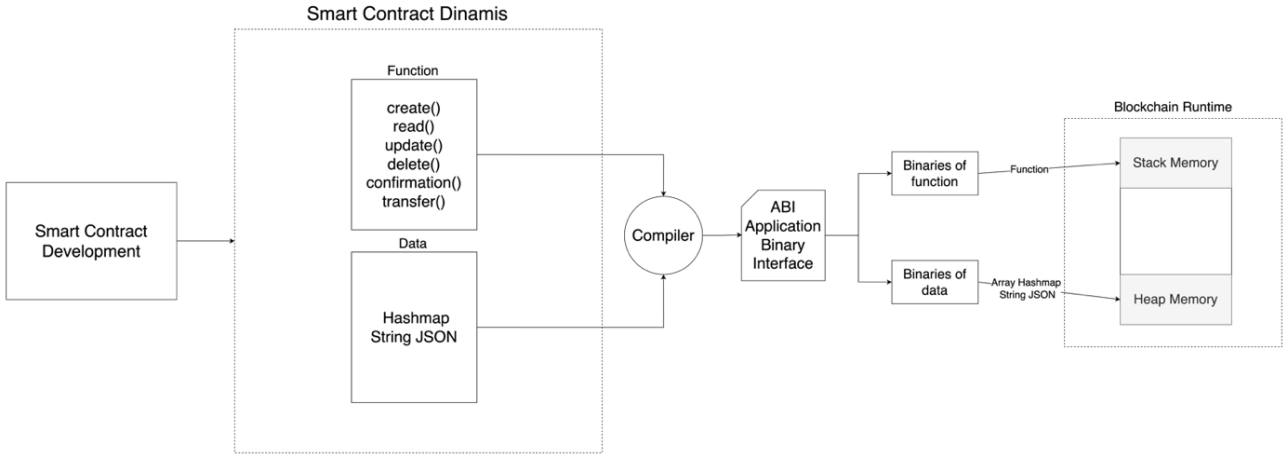

Fig. 7. Architecture Diagram of SCD's Data Structure

In conclusion, the nuanced architectural differences between SCS and SCD offer a revealing lens for their respective efficiency. The more dynamic and flexible approach of SCD demonstrated through its use of Hashmap String JSON and the elimination of unnecessary developmental steps, renders it a more adept choice for domains that require rapid adaptability. The targeted use of heap memory in SCD for data storage, as opposed to both stack and heap memory in SCS, further signifies its optimised approach for dynamic scenarios.As we further refined our analysis presented in Level 2, a significant innovation emerged in the form of parent–child smart contracts to add flexibility to our SCD model. The new structure was specifically designed to overcome the limitations observed in SCS, such as fixed parameters and the inability to adapt to new requirements. This approach allows a more fluid system capable of accommodating new commodities effortlessly. The intricate details of how key-value format data transitions through various stages, from mapping functions to parent and child smart contracts, and eventually to blockchain storage, are illustrated in Figure 8. Figure 8 not only serves as a technical deep dive but also shows the critical modifications made to the traditional SCS.

These changes primarily focus on revamping the key-value pair structure that governs data ownership. Unlike the previous system, where the SCS was designed to accommodate only predefined data formats typically suited for specific commodities such as fish or coffee, the updated model broadens its scope. Currently, the SCD can accept diverse data inputs, such as information about both fish and coffee, without requiring any structural reformatting. The SCD model offers an unprecedented level of flexibility, allowing real-time adjustments based on the user input. This marks a significant departure from the rigidly defined rules of the static model that are immutable once set. The dynamic nature of this new contract model not only enhances adaptability, but also enables the system to better cater to the fluctuating demands and conditions of the agricultural supply chain, particularly for commodities such as fish and coffee.

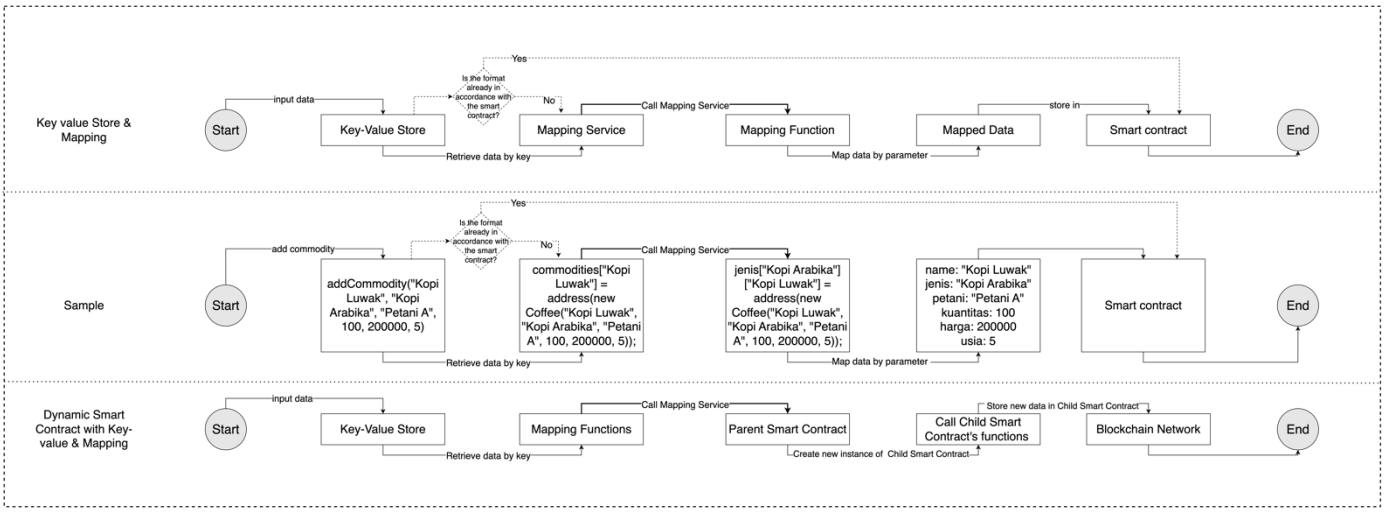

Fig. 8. Modification of SCS into Dynamic Ones Based on Key-Value Format

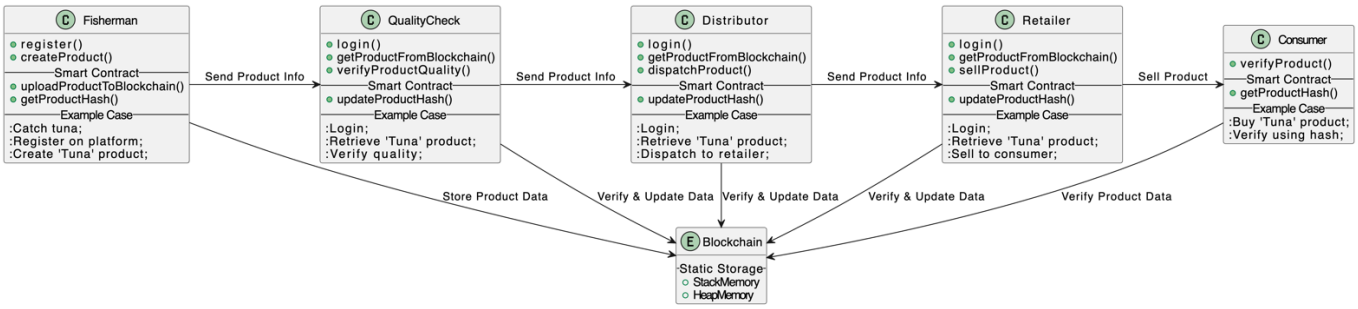

Fig. 9. SCS Flow Business

Building on the insights gathered from our Level 2 analysis, we now shift our focus to Level 3. This advanced level extends beyond the basic architecture and delves into the intricate technical aspects and practical applications of blockchain smart contracts. These are illustrated in Figures 8 and 9 for the two types of smart contracts that we consider SCS and SCD.

The smart contract stations and SCS are shown in Figure 9. This type of contract follows a static or a fixed framework. Although it operates effectively, this static nature imposes several limitations. Specifically, the SCS model uses a linear workflow that starts with a fisherman who catches the fish and ends with a consumer who purchases it. At each stage of the linear process, data must be verified and updated. This update cycle is defined by a set of static variables, such as the ID, name, quantity, and price of the product (in this case, the fish). Because the SCS is based on a rigid data structure, any change, such as

introducing a new type of fish into the supply chain, requires returning to earlier stages for re-verification and modification. This means that every time a change occurs, the smart contract must be revised or a new contract must be created. This rigidity can lead to inefficiencies and does not bode well for adapting to fluctuating market needs or complex dynamic supply chain requirements.

In contrast to SCS, the SCD illustrated in Figure 10 revolutionises the framework by offering unparalleled flexibility and dynamism. Using a Hashmap-String JSON data format, the system can accommodate a wide array of attributes, benefitting everyone in the supply chain from fishermen to consumers. Unlike SCS, where rigidity mandates frequent data updates and reverts to earlier stages for minor changes, SCD stands out for its modular approach, which enables fluid and dynamic data management.

Consider, for example, the complex task of managing multiple fish species in a supply chain. While SCS would necessitate a separate set of variables such as ID, name, quantity, and price for each species, SCD eliminates this need. A fisherman can effortlessly enter the details of various catches into the system using the flexible Hashmap String JSON format, without having to alter the existing smart contract. Consequently, quality assessors and distributors can move these products more efficiently along the supply chain, significantly reducing the time and computational costs.

The Key Takeaway One of the most notable advantages of SCD is its ability to streamline the entire data flow, negating the need for pathways between the stack and heap memory. This efficiency not only makes it suitable for complex and fast-changing scenarios but also makes it a more cost-effective choice for long-term blockchain implementation.

Therefore, our Level 3 analysis unmistakably demonstrates the edge SCD has in managing a dynamic and intricate supply chain like that of fisheries. This brings to the table an optimised, flexible, and scalable solution capable of adapting to fluctuating market conditions. This makes the SCD a benchmark in the realm of smart contract development and execution. The SCD is shown in Figure 10.

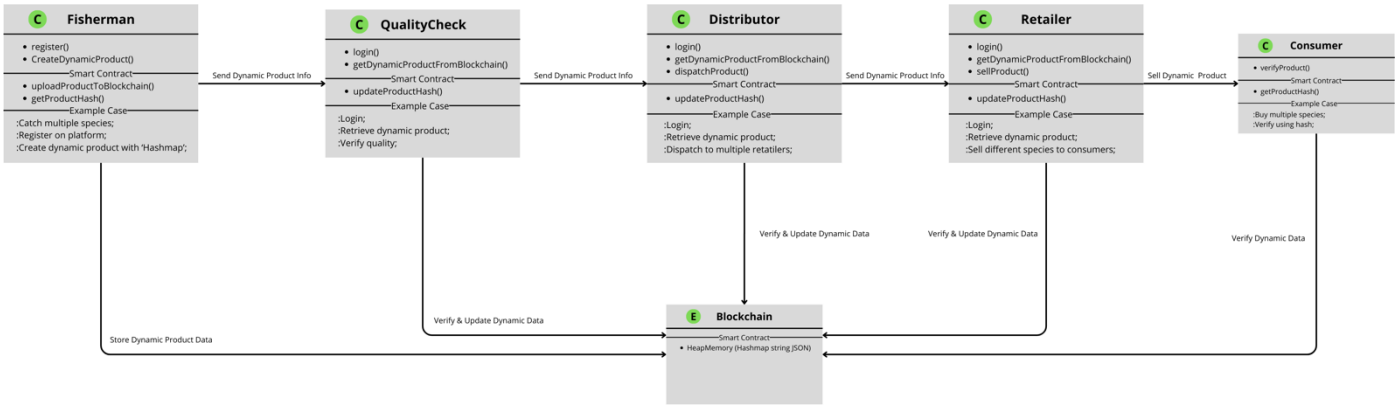

Fig. 10. SCD Flow Business

Building upon a more detailed diagram focusing on SCD, we explore additional layers of complexity that highlight its strengths. This advanced version of the diagram highlights a crucial aspect of the SCD's adaptability to dynamically add actors, actions, and their corresponding access rights to the system. Not only does it manage product data with exceptional flexibility, but it also extends this dynamism to role-based access control. This enables the seamless onboarding of new roles such as regulators, quality assessors, or even secondary distributors without necessitating any changes to the existing smart contract.

Additionally, the diagram introduces the concept of a 'Log Event', which captures all interactions and transactions occurring within the smart contract. This feature further enhances the traceability and accountability of the system, allowing real-time auditability of the supply chain. Stakeholders can effortlessly view changes, track product status, and verify actions, thereby bringing an unprecedented level of transparency to the ecosystem.

With these added layers, SCD not only streamlines data management but also offers modular and scalable role-based access control along with real-time tracking capabilities. This makes it an extraordinarily robust, transparent, and adaptable solution for intricate supply chain systems, clearly highlighting its superiority and setting a new standard for smart contract functionalities. A detailed diagram of the SCD is presented in Figure 11.

In Figure 12, our study unveils a pioneering prototype designed to bring about a sea change in how commodities are managed via SCD. Although the model is specially crafted with fish and coffee supply chains in mind, its architecture allows for effortless scalability and adaptation to diverse market conditions, thus creating a robust, versatile, and efficient contract management ecosystem.

As shown in Figure 12, the core of our proposed model consists of a Smart Contract for a Commodity Pool. This serves as the 'parent' smart contract from which specialised 'child' contracts, like the Fish Smart Contract and Coffee Smart Contract, are generated. The parent smart contract's attributes serve as a blueprint for its child contracts, ensuring uniformity while allowing for specific customisations. Within this framework, the application of SCD in the fields of fisheries and agriculture is demonstrated through a proof of concept. Here, the dynamic attributes of fishery products and agricultural commodities can be efficiently managed and updated in real-time, providing a responsive solution to the dynamic supply chain requirements in these sectors.

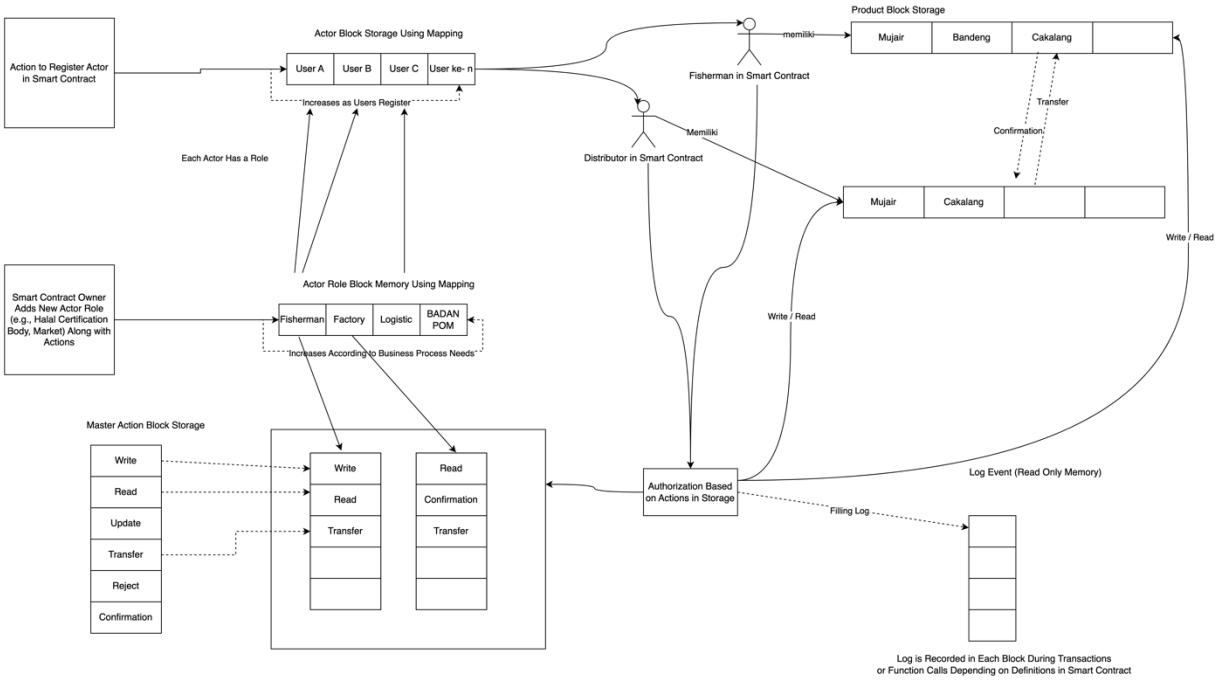

Fig. 11. SCD Detail Flow Business

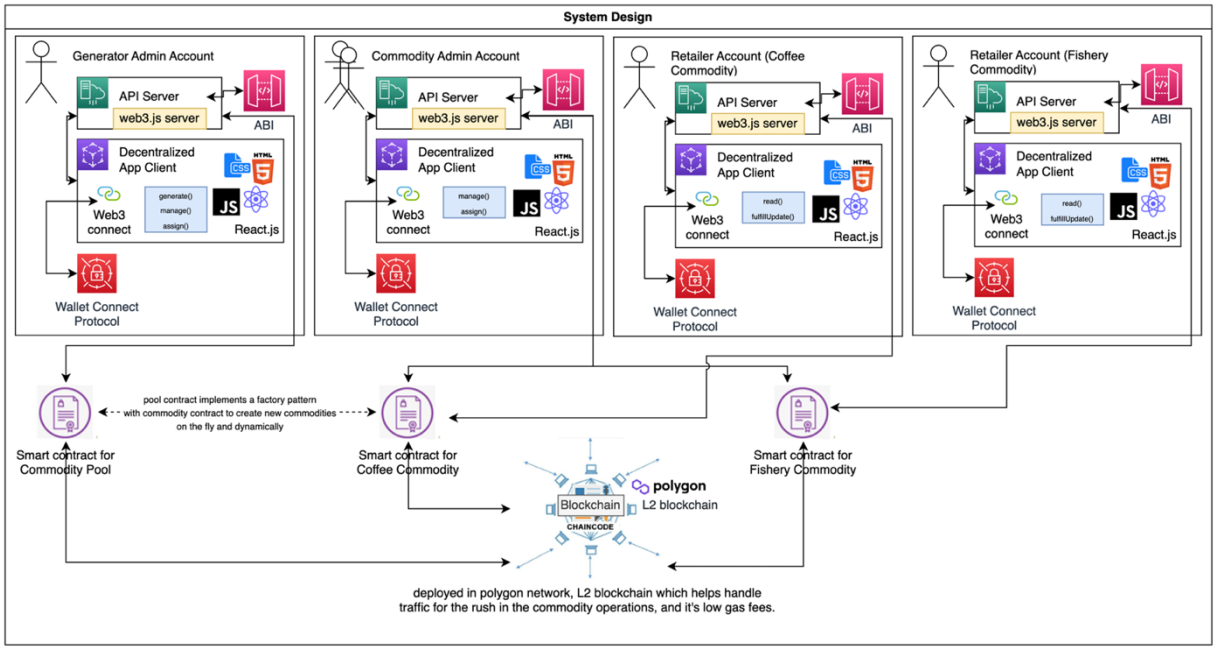

Fig. 12. SCD Proposed Model Design

In a static smart contract environment, transactions follow a template-based approach. Imagine you are working with a commodity like fish. The system strictly defines attributes such as type, weight, and origin. This inflexible structure constrains users to predefined fields, thereby limiting their adaptability and innovation. Once the values are input, they are transformed into key-value pairs that are further integrated into a comprehensive but rigid map data structure. This rigidity acts as a bottleneck, especially when adapting to new market requirements or when incorporating new types of data into the blockchain becomes necessary.

The Dynamic Smart Contracts (SCD) we offer largely function at the application layer of the blockchain. They bring about a certain degree of flexibility and adaptability without making any changes to the fundamental mechanics of the underlying Polygon network, such as consensus processes or block structures. The Solidity programming language was utilized in the development of these Smart Contracts for Agriculture (SCDs), which provide a flexible method for data management. This enables real-time updates and adjustments to

be made within the agricultural supply chain as required. In this particular methodology, when individuals engage with commodities, coffee is utilized as an illustrative instance, wherein they are not constrained by pre-established characteristics.

Alternatively, users have the flexibility to input a range of attributes, including but not limited to variety, weight, or origin, depending on their relevance to the transaction at hand. The customisable properties were transformed into key-value pairs and safely integrated into a versatile and dynamic map data structure. To ensure the integrity and security of smart contracts, various security measures are employed. These mechanisms encompass user authentication, data encryption, and stringent access controls. Their purpose is to thwart unwanted modifications to the key-value data structure. The flexibility possesses significance beyond its inherent characteristics, as it has the potential to significantly alter the current situation. This feature facilitates the process of duplicating and extending, hence supporting the inclusion of novel goods or characteristics with little complications. The blockchain incorporates a safe integration of a dynamic map consisting of key-value pairs, facilitating enhanced data retrieval and manipulation in a more user-friendly manner, as required.

In the context of these technological advancements, it becomes imperative to address the broader implications. It is crucial to consider the legal implications of utilizing SCD in a centralized industry structure. In this prototype, a centralized authority will have the highest access level to control modifications, additions, and reductions in access, reflecting the centralized power structure of the companies in the industry. This approach ensures that alterations to the smart contracts are legally compliant and consensual among involved parties.

Building on this foundation, the scalability and flexibility of our prototype, named the AniraBlock and illustrated in Figure 13, goes far beyond the fish and coffee sectors. The underlying architecture of the AniraBlock is designed to be universally adaptable, making it suitable for a multitude of industries, ranging from agriculture and land management to logistics, education, healthcare, finance, and energy sectors.

In summary, the AniraBlock serves as an example of the untapped potential of an SCD. By offering an innovative, highly scalable, and broadly applicable contractual management system, our model not only solves existing challenges but also equips businesses for future complexities. In doing so, it sets a groundbreaking standard poised to redefine smart contract implementation across sectors.

As shown in Figure 13, the AniraBlock is a tangible response to the limitations of the traditional SCS. It embodies the culmination of extensive research and development, encapsulating the dynamism and flexibility highlighted throughout this paper. This prototype serves as the bedrock on which multiple industry-specific smart contracts can be constructed, managed, and executed. It inherits the superior attributes of SCD and offers a holistic solution that paves the way for innovative, agile, and scalable blockchain implementation.

The AniraBlock model is not merely theoretical; it is a ready-to-deploy system that bridges the gap between static and dynamic contract management. By leveraging the versatility of SCD, AniraBlock sets a new benchmark for how contracts can be crafted, customised, and secured across a broad spectrum of industries.

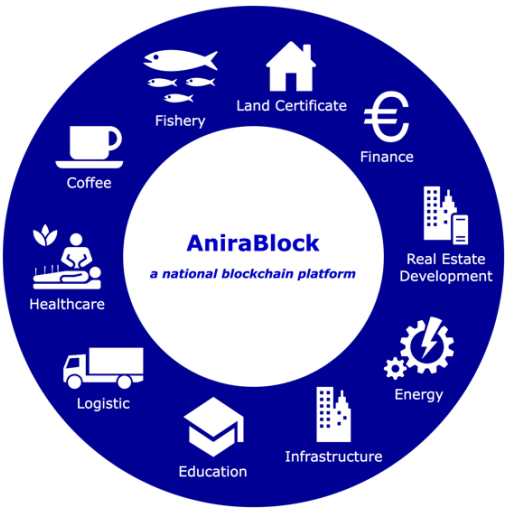

Fig. 13. AniraBlock Model Proposal

4. Conclusion

This study presents an innovative methodology for Smart Contracts known as the AniraBlock model, which surpasses the inherent constraints of SCS. The utilisation of SCD enables unparalleled adaptability and scalability, seamlessly accommodating the intricacies of real-world scenarios in several sectors including agriculture, logistics, healthcare, and finance. The fundamental basis of this innovation is the parent-child smart contract connection, which not only promotes adaptability but also facilitates the seamless integration of novel commodities and transaction types. This innovation signifies more than just a technological progression; rather, it signifies a profound transformation in the utilisation of smart contracts, transitioning from rigid predetermined structures to flexible and adaptable frameworks. This study, which is a significant contribution to the respective subject, establishes the foundation for a novel industry benchmark. This presentation highlights AniraBlock as an innovative solution that stimulates a more dynamic, effective, and inclusive blockchain environment, thus creating opportunities for future exploration and implementation in various industries.

References

1. K. Gopi, D. Mazumder, J. Sammut, N. Saintilan, Determining the provenance and authenticity of seafood: A review of current methodologies, Trends Food Sci. Technol. 91 (2019) 294–304.
2. K. Salah, M. H. U. Rehman, N. Nizamuddin, A. Al-Fuqaha, Blockchain for AI: Review and open research challenges, IEEE Access 7 (2019) 10127–10149.
3. Y. Kurniawan, M. A. Rizqulloh, Block cypher four implementation on field programmable gate array, Commun. Sci. Technol. 5(2) (2020) 53–64.

4. F. Branco, F. Moreira, J. Martins, M. Au-Yong-Oliveira, R. Gonçalves, Conceptual Approach for an Extension to a Mushroom Farm Distributed Process Control System: IoT and Blockchain, Adv. Intell. Syst. Comput. 930 (2019) 738–747.
5. R. Iqbal, T. A. Butt, Safe farming as a service of blockchain-based supply chain management for improved transparency, Clust. Comput. 23(3) (2020) 2139–2150.
6. N. Niknejad, W. Ismail, M. Bahari, R. Hendradi, A. Z. Salleh, Mapping the research trends on blockchain technology in food and agriculture industry: A bibliometric analysis, Environ. Technol. Innov. 21 (2021) 101272.
7. J. Førsvoll, S. Åndal, The application of blockchain technology for supply chain visibility-A case study of the fish farming industry, 2019.
8. P. K. Patro, R. Jayaraman, K. Salah, I. Yaqoob, Blockchain-Based Traceability for the Fishery Supply Chain, IEEE Access 10 (2022) 81134–81154.
9. C. Callinan, A. Vega, T. Clohessy, G. Heaslip, Blockchain Assimilation in Fisheries Supply Chain, blockchain.ieee.org, 2022.
10. M. Cordova, K. Nava-Aguirre, Achieving transparency through blockchain: sustainability of fishery supply chain management, Internext 2022.
11. E. Cruz, A. da Cruz, Using Blockchain to Implement Traceability on Fishery Value Chain., ICSOFT, 2020.
12. A. Dmitriev, K. Gjølsjø, A. Phillips, Fisheye on blockchain, 2022.
13. J. C. Ferreira, Fish Control Process and Traceability for Value Creation Using Blockchain Technology, Lect. Notes Netw. Syst. 419 (2022) 761–773.
14. X. Yang, D. Cao, J. Chen, Z. Xiao, A. Daowd, AI and IoT-based collaborative business ecosystem: a case in Chinese fish farming industry, Int. J. Technol. Manage. 82 (2020) 151-171.
15. S. Larissa, J. Parung, Designing supply chain models with blockchain technology in the fishing industry in Indonesia, IOP Conf. Ser. Mater. Sci. Eng. 1072 (2021) 012020.
16. N. I. Arvitrida, D. Rahmawati, D. Lastomo, Rindawati, Kusnadi, Fishery Supply Chains in Indonesia: Improvement Opportunities on The Downstream Side, Proc. Int. Conf. Ind. Enterp. Syst. Eng. (IcoIESE 2018), Yogyakarta, Indonesia, 2019.
17. T. M. Abebe, Blockchain Based Green Coffee Supply Chain Management to Improve Traceability and Transparency (Case Study on Sidama Coffee), Lect. Notes Inst. Comput. Sci. Soc.-Inform. Telecommun. Eng. LNICST 384 (2021) 304–318.
18. Blockchain Modeling for Traceability Information System in Supply Chain of Coffee Agroindustry, IEEE Conf. Publ., 2020.
19. A. J. Garcia Lozano et al., Decent work in fisheries: Current trends and key considerations for future research and policy, Mar. Policy 136 (2022) 104922.
20. M. van Keulen, J. Kirchherr, The implementation of the Circular Economy: Barriers and enablers in the coffee value chain, J. Clean. Prod. 281 (2021) 125033.
21. S. L. Bager, Blockchain is not a silver bullet for agro-food supply chain sustainability: Insights from a coffee case study, Curr. Res. Environ. Sustain. 4 (2022) 100163.
22. I. G. M. T. Pradana, Blockchain modelling for traceability information system in the supply chain of coffee agroindustry, Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2020 (2020) 217–224.
23. A. Tharatipyakul, Blockchain-Based Traceability System From the Users' Perspective: A Case Study of Thai Coffee Supply Chain, IEEE Access 10 (2022) 98783–98802.
24. V. Thiruchelvam, Blockchain-based technology in the coffee supply chain trade: Case of Burundi coffee, J. Telecommun. Electron. Comput. Eng. 10(3) (2018) 121–125.
25. T. Haldar, A. Damodaran, Identifying market power of retailers and processors: Evidence from the coffee supply chain in India, IIMB Manag. Rev. 34(3) (2022) 286–296.
26. A. Rijanto, Blockchain Technology Adoption in Supply Chain Finance, J. Theor. Appl. Electron. Commer. Res. 16(7) (2021) 3078–3098.
27. S. Liu, F. Mohsin, L. Xia, O. Seneviratne, strengthening smart contracts to handle unexpected situations, Proc. - IEEE Int. Conf. Decentralized Appl. Infrastruct. DAPPCON 2019 (2019) 182–187.
28. L. Chen, L. Xu, Z. Gao, Y. Lu, W. Shi, Protecting Early-Stage Proof-of-Work Based Public Blockchain, Proc. - Annu. IEEEIFIP Int. Conf. Dependable Syst. Netw. Workshop DSN-W 2018 (2018) 122–127.
29. A. Dolgui, D. Ivanov, S. Potryasaev, B. Sokolov, M. Ivanova, F. Werner, Blockchain-oriented dynamic modelling of smart contract design and execution in the supply chain, Int. J. Prod. Res. 58(7) (2020) 2184–2199.
30. F. Ghaffari, E. Bertin, N. Crespi, S. Behrad, J. Hatin, A Novel Access Control Method Via Smart Contracts for Internet-Based Service Provisioning, IEEE Access 9 (2021) 81253–81273.
31. C. S. Wright, Systems and Methods for Implementing Deterministic Finite Automata (DFA) via a Blockchain, Fourth Int. Congr. Inf. Commun. Technol. Springer Singapore, 2020, 499–512.
32. M. F. Afzaal, A. Rehman, S. Latif, N. A. Zafar, Blockchain based Automated Formal Model for Safety and Security in Smart Parking System, 2019 4th Int. Conf. Emerg. Trends Eng. Sci. Technol. ICEEST (2019) 1–6.
33. K. Sonbol, Ö. Özkasap, I. Al-Oqily, M. Aloqaily, EdgeKV: Decentralized, scalable, and consistent storage for the edge, J. Parallel Distrib. Comput. 144 (2020) 28–40.
34. Redis, Introduction to Redis, 2020.
35. B. K. Rai, S. Fatima, K. Satyarth, Patient-Centric Multichain Healthcare Record, Int. J. E-Health Med. Commun. 13(4) (2022) 1–14.
36. S. Kravenkit, C. So-In, Blockchain-Based Traceability System for Product Recall, IEEE Access 10 (2022) 95132–95150.
37. H. Yin, Y. Pan, H. Jiang, The Implementation of Simple Smart Contract Language and Its Compiler Based on Ethereum Platform, Proc. Int. Conf. Comput. Commun. Inf. Syst. ACM, Ho Chi Minh City Viet Nam, Aug. 2020, 77–82.
38. D. E. Smith, Innovating the Design Process: A Theatre Design Journey, A Focal Press book. Routledge, 2022.
39. Amazon Web Services, What Is a Key-Value Database, 2020.