Thesis for Master's Degree

Decentralized Herding: Improving Semi-Nomadic Livestock Supply Chain through Blockchain

Yesuilen Gulugdemberel

School of Electrical Engineering and Computer Science

Gwangju Institute of Science and Technology

2024

석 사 학 위 논 문

분산형 목축: 블록체인을 통한 반유목 가축 공급망 개선

굴루그뎀베렐 예수일렌

전 기 전 자 컴 퓨 터 공 학 부

광 주 과 학 기 술 원

2024

Decentralized Herding: Improving Semi-Nomadic Livestock Supply Chain through Blockchain

Advisor: Heung-No Lee

by

Yesuilen Gulugdemberel

School of Electrical Engineering and Computer Science

Gwangju Institute of Science and Technology

A thesis submitted to the faculty of the Gwangju Institute of Science and Technology in partial fulfillment of the requirements for the degree of Master of Science in the School of Electrical Engineering and Computer Science

Gwangju, Republic of Korea

Dec 13, 2023

Approved by

Professor Heung-No Lee

Committee Chair

Decentralized Herding: Improving Semi-Nomadic Livestock Supply Chain through Blockchain

Yesuilen Gulugdemberel

Accepted in partial fulfillment of the requirements for

the degree of Master of Science

Dec 13, 2023

Committee Chair ______________________
Prof. Heung-No Lee.

Committee Member ______________________
Prof. Brian J. d'Auriol.

Committee Member ______________________
Prof. Sunbeom So.

MS/EC 20211174 Yesuilen Gulugdemberel. Decentralized Herding: Improving Semi-Nomadic Livestock Supply Chain through Blockchain. School of Electrical Engineering and Computer Science. 2024. 54p. Advisor: Prof. Heung-No Lee.

Abstract

This thesis, "Decentralized Herding: Improving Semi-Nomadic Livestock Supply Chain through Blockchain," presents an innovative approach to addressing the challenges faced by semi-nomadic livestock supply chains, exemplified through Mongolia's case. It explores the potential of blockchain technology to enhance traceability, data integrity, and overall supply chain management, thereby tackling issues like overgrazing and food safety. The study emphasizes the importance of user-centric design and regulatory compliance in blockchain applications. Through this exploration, the thesis offers strategies to improve livestock productivity and establish equitable pricing mechanisms, providing valuable insights for similar supply chains worldwide.

MS/EC 20211174 굴루그뎀베렐 예수일렌. 분산형 목축: 블록체인을 통한 반유목 가축 공급망 개선. 전기전자컴퓨터공학부. 2024. 54p. 지도교수: 이 흥 노.

국 문 요 약

이 논문 ”분산형 목축: 블록체인을 통한 반유목 가축 공급망 개선”은 몽골의 사례를 통해 예시된 반유목 가축 공급망이 직면한 문제를 해결하기 위한 혁신적인 접근 방식을 제시합니다. 추적성, 데이터 무결성 및 전반적인 공급망 관리를 향상시켜 과잉 방목 및 식품 안전과 같은 문제를 해결하는 블록체인 기술의 잠재력을 탐구합니다. 이 연구는 블록체인 애플리케이션에서 사용자 중심 설계와 규제 준수의 중요성을 강조합니다. 이러한 탐구를 통해 논문은 가축 생산성을 향상시키고 공평한 가격 책정 메커니즘을 확립하기 위한 전략을 제공하여 전 세계 유사한 공급망에 대한 귀중한 통찰력을 제공합니다.

Contents

List of Tables

List of Figures

Chapter 1

Introduction to Blockchain Technology and Livestock Supply Chain Management

1.1 Introduction

Blockchain technology is revolutionizing various industries by offering a secure, transparent, and decentralized means of recording transactions and managing data. In parallel, the livestock supply chain represents a critical component of agriculture, encompassing the complex processes involved in the production, processing, and distribution of livestock and livestock-derived products. This chapter serves as an introduction to the convergence of these two domains, exploring the potential of blockchain technology to address the unique challenges within the Mongolian livestock supply chain.

1.2 Motivation

Livestock husbandry, an integral facet of agriculture, encompasses a wide array of practices crucial for food, fiber, transportation, and more. While modern technology has significantly impacted sedentary farming, which represents the majority of livestock production, nomadic herding remains comparatively underexplored.

Mongolia, my homeland, is one of the few semi-nomadic herding nations. However, there is limited research on integrating innovative technologies like blockchain into our herding supply chain. This gap presents a unique opportunity to leverage the knowledge and skills I've acquired to address the challenges faced by this sector.

My motivation for this thesis stems from a deep-rooted connection to Mongolia's semi-nomadic herding heritage. By embracing blockchain technology, we can potentially overcome challenges such as overgrazing and supply chain opacity. This research not only aims to benefit Mongolian herders but also contributes valuable insights to the global livestock and blockchain community.

In essence, this thesis endeavors to blend tradition with technology, envisioning a brighter and more sustainable future for semi-nomadic herding practices worldwide.

1.3 Outline of this Thesis

The thesis is structured into four chapters, each contributing to our overarching objectives. Chapter one provides an introductory overview, delving into the foundational concepts of blockchain technology and supply chain management, with a particular emphasis on the distinct challenges confronting the Mongolian livestock supply chain. Chapter two encompasses a comprehensive review of existing literature, presenting diverse blockchain-based livestock chain management systems, elucidating their enhancements with traditional systems, and addressing gaps prevalent in the current research landscape. In chapter three, we introduce an innovative livestock supply chain management system meticulously designed to contend with the unique challenges characteristic of the Mongolian livestock supply chain. This proposal is informed by our analysis of extant literature and the identification of research gap. Finally, in chapter four, we present the outcomes stemming from the implementation of this system on the Ethereum(WorldLand) network and conduct a rigorous assessment of its performance against the previously delineated system requirements. Furthermore, chapter four serves as the conclusive segment of the thesis, offering insights into potential avenues for future research endeavors within this domain.

1.4 Methodology

The research methodology encompasses several key stages as shown in Figure 1.1. Initially, a comprehensive review of relevant literature and statistical reports on Blockchain technology, supply chain management, and nomadic livestock herding is conducted to gather essential data and identify the challenges. Subsequently, the identified issues within the nomadic livestock supply chain are evaluated in the second stage. The third phase involves assessing the potential of Blockchain technology within the livestock industry supply chain. Following this, the research proceeds to map how Blockchain can effectively address the challenges specific to the livestock supply chain in the context of nomadic herding in Mongolia. Finally, the last step entails evaluating the practical application and presenting recommendations for the livestock supply chain model within the nomadic herding livestock sector.

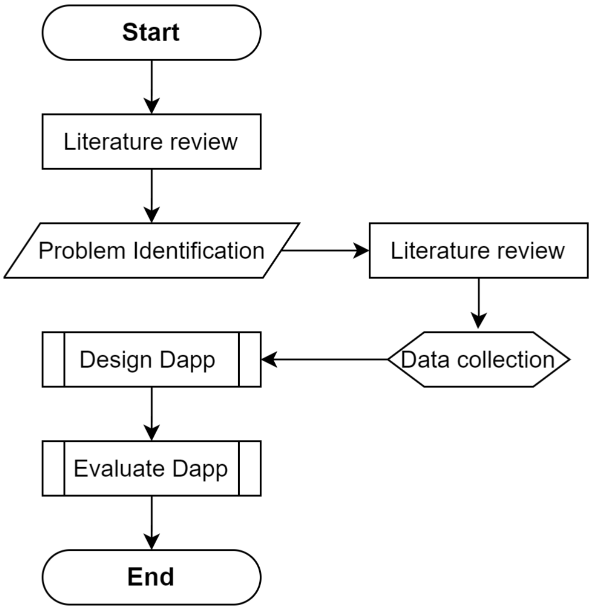

Figure 1.1: Research Flow.

1.5 Blockchain Technology

As per the insights of Don and Alex Tapscott [1], blockchain is a decentralized and transparent digital ledger technology that serves as a tamper-proof system for recording economic transactions. Beyond its initial role in recording financial transactions, blockchain has evolved to manage various data types. The launch of the Bitcoin network in 2008 marked a key moment, initiating the era of public digital ledgers. Since then, blockchain has expanded into diverse fields such as digital identity, government voting, financial services, supply chain management, and media, leading to the development of numerous blockchain networks beyond Bitcoin. Some notable examples include Ethereum, Hyperledger, Ripple, and R3 Corda [2]. Despite their unique features, these networks share a common foundational architecture.

According to Ismail and Materwala [3], the blockchain architecture is segmented into four layers: infrastructure, platform, distributed computing, and application. The infrastructure layer encompasses the hardware components essential for running the blockchain network, including nodes, storage, and network facilities. Blockchain technology boasts several key features that render it valuable [4] including decentralization, immutability, transparency, and trustlessness. The precise architecture and design of a blockchain can vary, contingent on the specific network and its intended use case.

1.5.1 Bitcoin Network

Bitcoin, launched in 2008 by an individual or group known as Satoshi Nakamoto, stands as the first and most prominent cryptocurrency, marking a significant shift in digital currency and blockchain technology. Key characteristics of Bitcoin include:

- Digital Currency: It is a digital or virtual currency without any physical form, secured using cryptographic techniques.
- Mining: This process not only creates new bitcoins but also adds transactions to the blockchain, rewarding miners with new bitcoins and transaction fees.
- Pseudonymity: Transactions are pseudonymous, recorded on the public blockchain with cryptographic addresses that represent ownership but do not directly reveal the user's identity.

1.5.2 Ethereum Network

Ethereum, introduced by Vitalik Buterin in 2013 and launched in 2015, stands apart from Bitcoin with its emphasis on programmability. It allows the creation of decentralized applications (dApps) and the execution of smart contracts, significantly advancing blockchain technology [5]. Its key features include:

- Smart Contracts: Self-executing contracts that automate execution upon certain conditions, eliminating intermediaries.
- Ether (ETH): Ethereum's native cryptocurrency, used for transaction fees and computational services, and incentivizing miners.
- EVM (Ethereum Virtual Machine): A platform for executing smart contracts and transactions, supporting multiple programming languages.
- Gas: A measure of computational work, with users paying fees in Ether for transaction processing.

Ethereum has evolved into Ethereum 2.0, enhancing scalability, security, and sustainability with Proof of Stake (PoS) and sharding, reinforcing its position in the blockchain ecosystem.

1.5.3 Hyperledger Fabric

Hyperledger Fabric, part of the Linux Foundation's Hyperledger project, is an open-source, permissioned blockchain framework designed for building secure, scalable, and customizable enterprise and consortium blockchain networks [3]. Key features of Fabric include:

- Permissioned Network: Tailored for controlled access, offering privacy and suitability for enterprise use.

- Modularity: Its architecture allows customization of consensus algorithms, membership services, and smart contract engines.
- Chaincode: Hyperledger uses "chaincode" for smart contracts, supporting languages like Go, JavaScript, and Java, adding to its flexibility.

1.5.4 R3 Corda

R3 Corda, launched in 2016 by the R3 consortium, is an open-source blockchain platform designed for business and enterprise applications, particularly in private and consortium networks. Its key features are:

- Smart Contracts: It facilitates smart contract creation and implementation with "CorDapps," using Java or Kotlin for developer accessibility.
- Pluggable Consensus: Corda's consensus mechanism is adaptable, enabling participants to choose algorithms fitting their specific needs.
- Legal Framework: The platform includes the "Corda Legal Identity Toolkit" for legal identification on the blockchain, crucial in heavily regulated industries.

1.5.5 Decentralized application (Dapp)

Decentralized applications, commonly referred to as dApps, represent a crucial component of the blockchain ecosystem, notably popularized by Ethereum. DApps are essentially software applications built on blockchain technology, utilizing smart contracts as their backbone. Unlike traditional apps, dApps operate on a decentralized network of computers, making them resistant to censorship and fraud while ensuring transparency and security. Ethereum's pioneering role in dApp development has led to the creation of a wide range of applications beyond cryptocurrencies. These include decentralized finance (DeFi) platforms, non-fungible token (NFT) marketplaces, supply chain management systems, and more. DApps facilitate peer-to-peer interactions, enabling users to transact, exchange assets, or engage in various activities without relying on intermediaries.

One of the key advantages of dApps is their open-source nature, encouraging collaborative development and innovation within the blockchain community. They also often incorporate governance mechanisms that allow users to participate in decision-making processes, giving them a sense of ownership and control over the application's direction. However, dApps are not without their challenges. Scalability, user-friendliness, and regulatory compliance remain ongoing concerns. Nevertheless, the continuous evolution of blockchain technology and dApp development frameworks promises to address these issues, further expanding the potential applications of decentralized applications in various industries.

1.6 Livestock Supply Chain Management

The livestock supply chain is a comprehensive system that includes the production, processing, and distribution of livestock and their derived products, such as meat, dairy, and wool. This chain incorporates diverse livestock-rearing practices, tailored to various geographical and cultural contexts:

- **Farming**: This is the predominant method globally, particularly in developed countries. It includes both large-scale, industrial farming operations and small-scale, traditional farming methods.

- **Nomadic and Semi-Nomadic**: These methods are more prevalent in areas conducive to pastoralism, like parts of Africa, Central Asia, and the Middle East. Their occurrence has diminished somewhat due to the rise of modern agricultural practices and the effects of urbanization.

The supply chain comprises several key participants [6], including production materials suppliers who provide essential resources such as feed and veterinary medicine, farmers responsible for the rearing and breeding of livestock, livestock slaughtering and processing enterprises engaged in the processing of livestock for subsequent distribution, vendors such as wholesalers, retailers, and supermarkets that facilitate the distribution of livestock products, and consumers who represent the ultimate end-users of these products. Each stage in the livestock supply chain assumes a critical role in preserving the quality and safety of livestock products throughout the entire process.

1.6.1 Mongolian livestock supply chain

Agriculture, specifically livestock husbandry, holds paramount significance within Mongolia's economy. As delineated in the 2022 report by the National Statistical Office of Mongolia (NSA) [7], the livestock industry contributes a substantial 84.2 percent to the Gross Agricultural Products of the nation. Remarkably, this sector serves as the sole income source for 18.5 percent of households across the country. The Mongolian livestock sector possesses a distinct and advantageous feature due to its nomadic herding traditions spread across vast pasturelands, covering approximately 110 million hectares. This geographical advantage, situated in proximity to the world's second-largest economy, China, enables the maintenance of a substantial, organically grass-fed livestock production system. Despite its pivotal role in both the economy and the livelihoods of the population, the structure of the Mongolian livestock supply chain has remained relatively unaltered. Figure 1 illustrates the Mongolian livestock supply chain, as derived from a survey [8] conducted by the United Nations Conference on Trade and Development (UNCTAD).

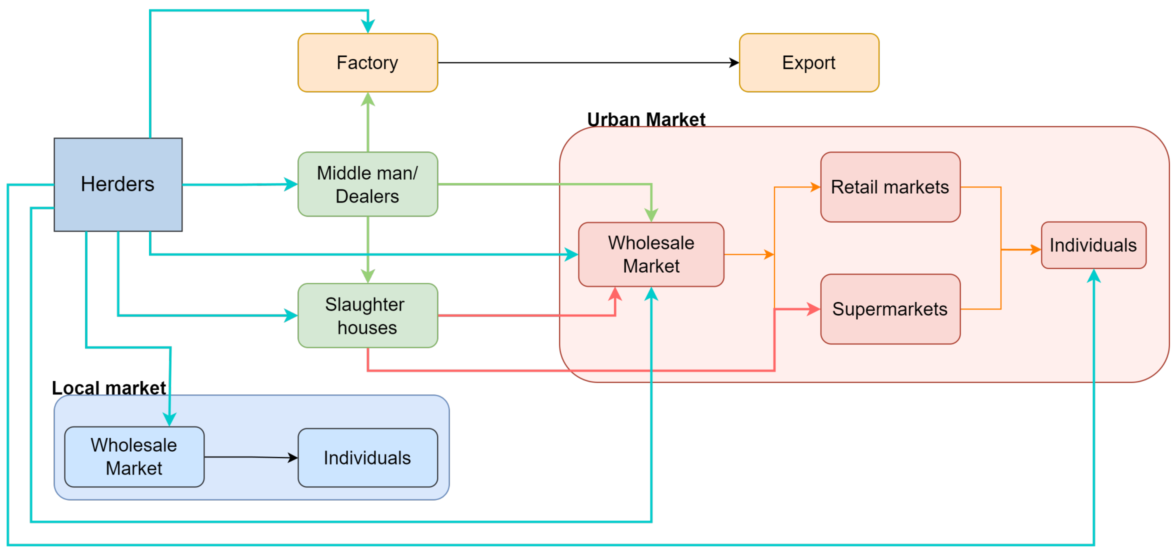

Figure 1.2: Mongolian livestock product supply chain diagram

Within the Mongolian context, the livestock product supply chain is a multi-tiered process commencing with herders functioning as primary producers. Subsequently, the supply chain encompasses intermediaries, slaughterhouses, meat processing facilities, and dairy factories. Ultimately, the resulting meat and dairy products find their way into domestic markets through a network of food markets and supermarkets. Additionally, these products are exported to international markets, contributing to the global reach of Mongolia's livestock industry. Despite the substantial potential, Mongolia's meat exports remain at less than 10 percent of their capacity [9], largely due to the predominant practice of herder-based slaughter, which accounts for 97 percent of meat processing. The Mongolian livestock supply chain, characterized by its uniqueness, presents specific challenges not commonly encountered in other countries. These distinctive challenges will be discussed in detail in the following section.

1.6.2 Challenges of Mongolian Livestock Supply Chain Management

The Mongolian livestock supply chain faces critical challenges common to the broader food industry, such as food safety, traceability, labor fairness, and pricing equity. However, it confronts distinctive hurdles rooted in overgrazing, food safety apprehensions, and unsustainable price differentials. Research [10] by the International Monetary Fund (IMF) illustrates a threefold surge in the livestock population, reaching 70 million since 1990, surpassing the land's carrying capacity. This imbalance results in significant livestock losses during severe cold spells, amounting to approximately 12 percent of the country's GDP. Consequently, it propels a rural-to-urban migration trend, with approximately one in four Mongolians residing in the ger district on the outskirts of Ulaanbaatar, which strains local infrastructure and exacerbates air pollution. Furthermore, land degradation spawns yellow dust storms[11], impacting both public health and regional economies, affecting not only Mongolia but also neighboring East Asian nations, including China, Japan, and South Korea. Furthermore, we delve into the key challenges that require systematic addressing within the Mongolian livestock supply chain:

- Overgrazing: It is a term used in agriculture and ecology to describe a situation where livestock, such as cattle, sheep, or goats, consume vegetation in an area at a rate that exceeds the natural capacity of the land to regenerate and regrow the plants. In Mongolia, despite having vast pasturelands covering approximately 110 million hectares, overgrazing has led to significant pasture degradation, with an average overstocking rate of 2.3 times beyond the current carrying capacity [9]. This degradation, affecting nearly 65 percent of rangelands, poses threats to both the environment and livestock quality, making the livestock sector highly vulnerable to climate change and extreme weather events.

- Food health: The dynamic interaction between Mongolian herders and their livestock entails inherent risks. The extensive grazing lands in Mongolia, where livestock roam freely, expose them to various diseases originating from wildlife. Notably, foot-and-mouth disease has garnered international attention due to its viral nature, localized impact, and non-zoonotic nature; this disease presents a prominent challenge for Mongolian meat exporters. The most recent outbreak of foot-and-mouth disease occurred in May 2020. Additionally, the lack of traceability and transparency in the supply chain poses disadvantages to end consumers. This opacity raises concerns about the origin of products, their safety for consumption, and whether appropriate sanitary and health measures were taken during meat processing.

- Product quality: Enhancing livestock productivity constitutes a pivotal factor in attaining competitiveness within the sector. The World Bank's Livestock Sector Study has revealed that Mongolian beef carcasses weigh less than half of their international counterparts, ranging from 120 kg to 130 kg compared to the global standard of 250 kg to 300 kg. Furthermore, the average cow milk yield witnessed a 19 percent decrease between 2010 and 2015. Numerous factors underlie this issue of low and diminishing productivity, encompassing malnutrition, animal diseases, natural risks, and climate-related hazards.

- Unfair price: Middlemen are vital components of the supply chain, as depicted in Figure 1.1, they play a crucial role in establishing and sustaining relationships with factors and customers due to their access to current market insights acquired from extensive business networks. Survey results suggest that middlemen primarily serve as essential intermediaries bridging the gap in the market by connecting isolated herders with potential buyers, as herders face constraints in transporting their animals to processors or markets. However, the lack of transparency and information contributes to substantial price differentials between the selling prices negotiated by herders and the buying prices paid by end customers.

In this chapter, we introduced blockchain technology and its relevance to the study. We also provided an overview of the Mongolian livestock supply chain, highlighting key challenges such as overgrazing, food safety, declining product quality, and pricing disparities. These insights set the stage for the subsequent chapters, where we explore the integration of blockchain into Mongolia's livestock supply chain to address these challenges. One potential solution to these challenges lies in the application of blockchain technology. Blockchain can enhance transparency and traceability in the supply chain, helping to ensure the safety of livestock products. It can also facilitate better data management and sharing among stakeholders, potentially leading to improved livestock productivity. Moreover, the use of smart contracts on the blockchain can help in establishing fair pricing mechanisms and reducing the influence of intermediaries, benefitting both herders and consumers.

In the next chapter, we will delve deeper into the relevant literature, examining similar studies and frameworks to inform our research methodology and approach. We will also explore how blockchain technology has been applied in other agricultural and supply chain contexts, drawing lessons that can be adapted to the unique challenges of Mongolia's nomadic herding system.

Chapter 2

Related works of action recognition

In this chapter, we will conduct a thorough literature review focusing on the agricultural supply chain, with a particular emphasis on the livestock sector. Our objective is to identify existing research gaps within this domain, which will subsequently guide the formulation of our research contributions. Blockchain technology has witnessed rapid advancement, leading to its extensive adoption and extensive exploration across various sectors. Supply chain management has emerged as a prominent domain for harnessing the potential of blockchain technology, with particular emphasis on its application in the agricultural sector. However, within the specialized realm of livestock supply chains, the literature remains comparatively scarce in contrast to the broader agricultural supply chain landscape.

2.1 Literature review

Recent literature [12] sheds light on the challenges associated with the adoption of Blockchain Technology (BCT) within agricultural supply chains. It underscores the necessity for further research into the factors influencing this adoption and the scalability challenges inherent in BCT protocols. Notably, this study accentuates the potential benefits of BCT in precision agriculture, smart farming, and bolstering supply chain resilience, while also recognizing its positive impact on transparency and food safety. In the context of livestock traceability in Indonesia, [13] places a spotlight on the importance of providing clear and user-friendly information while stressing the involvement of government entities. Additionally, it underscores the favorable perception of blockchain technology among consumers.

The integration of blockchain and the Internet of Things (IoT) in e-agriculture is the focal point of [14], offering a proposal for a blockchain-IoT architecture geared towards data management within the agricultural realm. This integration stands to fortify the connections within the value chain, enhance data-related processes, and cultivate trust among stakeholders. It is imperative to note that empirical validation of this proposed architecture remains a pertinent area of future exploration.

Furthermore, research by [15] delves into the integration of blockchain technology into Sri Lankan livestock farming, with a specific emphasis on augmenting scalability, security, and performance. This endeavor aims to address the pressing need for enhanced data management, transparency, and trust within the sector. The study examines the effective integration of blockchain, evaluates potential advantages, and identifies implementation challenges and constraints. It draws upon a myriad of data sources, including IoT sensors, RFID tags, Hyperledger Fabric, and REST APIs, ultimately presenting a permissioned blockchain architecture featuring smart contracts to bolster data security and transparency in the livestock industry. While promising, this study acknowledges the complexity of the proposed system and underscores the significance of user-friendly applications, along with the need to address potential LAN network security concerns for future refinements.

Moreover, [16] takes a focused approach by developing a blockchain-based agricultural service platform that capitalizes on smart contracts and IoT technology to optimize drone-based plant protection services. This initiative seeks to address challenges encompassing service traceability, fraud prevention, information security, and privacy within platform-based agricultural services. The research introduces a framework underpinned by smart agricultural service platform integration of IoT and blockchain technologies. It offers insights into the intricate nature of designing smart contract-based agricultural services, emphasizing the importance of optimization techniques to accommodate multiple service requests. Additionally, the study underscores the efficiency, security, and applicability of the designed smart contract while acknowledging limitations concerning fixed base stations for drones and the scope of disruption events in agricultural service optimization. These aspects beckon further research opportunities within this domain.

Moreover, reference [17] tackles challenges present within agro-industrial supply chains in Indonesia and explores blockchain technology's potential to enhance efficiency and transparency. The study delineates key challenges inherent in these supply chains, investigates the applications of blockchain, and deliberates on the technology's prospective role in the future of Indonesian agro-industry supply chains. It proffers a model illustrating how blockchain can ameliorate these supply chains, while also highlighting the challenges and unresolved matters in its implementation. These include the imperative need to address accessibility, governance, regulatory, technical, and digital literacy issues among farmers. Notwithstanding these challenges, blockchain technology holds promise as a salient solution to the predicaments faced by agro-industrial supply chains in Indonesia, with the proposed model serving as a foundational framework contingent upon the resolution of regulatory and implementation concerns.

Furthermore, [18] employs an exploratory case study approach, utilizing the Smart Trade Networks (STN) Proof of Authority (PoA) blockchain system to track assets and gather data. The results underscore the potential benefits of this approach, notably the heightened transparency, information sharing, and operational efficiency achieved within the supply chain. Theoretical contributions extend to the realm of supply chain governance, the role of blockchain in digital transformation, and the harmonization of supply chain and information system strategies. Meanwhile, practical implications revolve around addressing digital disruption, ensuring interoperability, and substantiating technical feasibility, and economic viability. The study does, however, grapple with limitations such as manual data collection and the imperative need for industry-oriented research and trust-related investigations within blockchain systems.

Finally, [19] aspires to construct a dynamic supervision model for the rice supply chain through the deployment of blockchain and smart contracts. The primary goal is to enable real-time management, encompassing various aspects, from business information to hazard information and personnel data, with a view to enhancing rice quality and safety. The research leverages diverse IoT devices, including RFID, NFC, mobile phones, computers, and GPS, to source data. It orchestrates a dynamic supervision model framework fortified by blockchain and smart contracts, complete with three custom-designed smart contract types. The study amalgamates asymmetric encryption, virtual regret minimization algorithms, and multisource heterogeneous fusion algorithms. The outcomes are evident in the form of a prototype system, affirming the model's capacity to facilitate real-time management across the rice supply chain's lifecycle. Nevertheless, the study is upfront about the operational constraints associated with blockchain computing and storage resources, which affect operational speed and storage time. Furthermore, the research predominantly centers on the rice supply chain, leaving scope for adaptation to other food crops. Overall, it proffers a pragmatic solution geared towards digitizing and supervising food supply chains to fortify food safety and quality. Future research pathways may involve transposing similar models to alternative food crops and surmounting computing resource limitations.

In conclusion, the literature showcases a burgeoning interest in the application of blockchain technology within the realm of supply chain management, particularly in the agricultural and livestock sectors. The studies elucidate the multifaceted benefits and challenges associated with the integration of blockchain, ranging from enhanced transparency, security, and operational efficiency to complex issues like scalability and regulatory hurdles. While promising, these endeavors underscore the need for continuous empirical validation, user-friendly applications, and resolution of technical constraints. Consequently, these studies provide invaluable insights into the evolving landscape of blockchain technology in agricultural supply chains, paving the way for our research and industry adoption. The comparative analysis of the related works in this field discussed so far, along with our proposed approach has been summarized in Table 1.

	Analyzed Characteristics			
Ref.	**Objective**	**Key Findings**	**Tech**	**Limitations**
[7]	Real-time tracking of livestock supply chain based on Blockchain-IoT	Supply chain transparency and commodity trust using RFID electric tag and electronic tag	RFID, electronic tag	Too expensive for farm use
[15]	Enhancing scalability, tracking, and transparency of Sri Lankan livestock farming	Permissioned blockchain architecture with smart contracts. Acknowledgment of system complexity and security concerns	Hyper ledger, IoT sensors	User-friendliness, LAN network security challenges
[16]	Optimization of drone-based plant protection services against fraud and privacy issues	Comprehensive framework and an efficient execution procedure for agricultural services with IoT and blockchain integration	Ethereum, IoT-drone, GPS sensor	limitations for fixed base stations and disruption events
[17]	Blockchain-based architecture for Indonesian agro-industry's quality assurance and food safety	A thorough examination of the Indonesian agro-industry existing challenges, and proposing a blockchain-based model to solve these issues.	Block chain, IoT RFID, sensors	Accessibility, governance, regulation, and technical issues need to be addressed.
[18]	Design blockchain based multi signature approach for the Australian beef industry	Beef supply chain governance via relational governance and improving transparency and efficiency.	Ethereum, magic link, IoT, sensors	Manual data gathering, industry-focused research
[14]	Integration of blockchain and IoT in e-agriculture to enhance data management	Strengthened value chain linkages, improved data management, and enhances resource utilization for sustainable farming	Block chain, IoT	Empirical validation to assess its practical implementation
[19]	Construction of a dynamic supervision model for the rice supply chain	Real-time management of the rice supply chain, improved data security, and interconnectivity	Hyper ledger, Cloud database	Computing resource and speed limitations
[20]	Enhancing traceability in livestock-based products	Comprehensive comparison of tracking technology options	RFID, DNA Bar coding	limited adoption capacity, coordination
NLS	Blockchain solution for food Health, Overgrazing, and Transparency Concerns	Dapp architecture and algorithm for calculating meat grading for semi-nomadic livestock supplychain	Ethereum, ear tag, QR code	Meat grading equation improvement, data storage solution

Table 2.1: Summary of Literature Review

2.2 Research Gap

Blockchain technology has made substantial advancements in a multitude of domains, spanning finance, entertainment, art, and notably, supply chain management. The supply chain sector has emerged as a focal point of interest within the blockchain community, with applications proliferating across diverse industries such as food, merchandise, natural resources, and various research domains. These developments have captured the scholarly community's keen attention.

This thesis narrows its research focus to the food supply chain, with a specific emphasis on the livestock sector. While existing literature has addressed challenges within the livestock supply chain and proposed innovative solutions to address these global challenges, the impact of blockchain technology in this context, especially concerning farming practices, has been explored in prior research.

Nevertheless, a noteworthy research gap persists. To date, there is a conspicuous absence of studies examining the factors contributing to issues such as overgrazing and overpopulation of livestock within nomadic herding conditions. This research endeavor seeks to address this void by conducting a comprehensive analysis of the challenges inherent in nomadic herding livestock supply chains. Moreover, the study endeavors to illuminate the ways in which these challenges affect both the climate and public health, providing invaluable insights into this vital yet underexplored facet of the livestock industry.

2.3 Thesis contribution

In light of blockchain's potential to impact livestock management and agricultural practices, this research seeks to address a notable gap within the academic discourse. It undertakes the important task of conducting a comprehensive investigation into the complex factors contributing to overgrazing and livestock overpopulation in nomadic herding contexts. This effort results in the introduction of an innovative supply chain framework leveraging blockchain technology to systematically address these challenges. By venturing into unexplored territory within the livestock supply chain domain, this thesis aims to enhance our understanding of this intricate field and propose practical solutions for the ecological and public health issues associated with nomadic herding practices. In summary, this research represents a significant step towards promoting sustainable livestock management and bolstering global food security in a thoughtful manner.

Chapter 3

NLS Dapp Smart Contract and Architecture Development

In this chapter, we will outline the prerequisites for our Nomadic Livestock Supply Chain (NLS) decentralized application (Dapp) in light of the challenges elucidated in Chapter 1. Building upon our analysis of relevant literature in the field, we will propose solutions to address these challenges. Subsequently, we will embark on the design and development of our NLS Dapp Smart contract to align with these proposed solutions. Following this, we will advance to the next chapter, where we will undertake the development, implementation, evaluation, and discussion of our Dapp, encompassing its accomplishments and constraints.

3.1 NLS Dapp Smart Contract Creation

In order to ascertain the technical and workflow prerequisites for the Dapp, we have compiled Table 3.1, associating the issues we intend to address with their respective solutions, drawing upon insights from prior analogous endeavors. This process has enabled us to articulate the technical requisites necessary to fulfill each corresponding solution.

As previously highlighted, the core innovation of our research centers around the

Table 3.1: NLS Dapp requirement

Problem	**Solution**	**Technical approach**
Food Health	As per references [12][13][14][21], ensuring food health relies on accurate tracing of origin, ingredients, and processing methods, enabling informed consumer choices and swift outbreak response.	Traceability
Product Quality	In line with [14][15][18], Maintaining product quality requires producer compliance with standards, rigorous inspections, and transparent disclosure, facilitated by an immutable ledger.	Immutable ledger
Unfair price	According to [15][16][17][18], achieving fair pricing and equitable labor conditions relies on complete supply chain transparency, spanning from the production stage to the end customer	Transparency
Overgrazing	Political decisions, like implementing pasture taxes and aligning product quality with consumer demand, offer potential solutions to address overgrazing, as suggested by [9][10][22]	Quality grading system

introduction of a smart contract-based solution aimed at mitigating the issue of overgrazing. Research [8] has shown that overgrazing leads to a decline in livestock product quality, and given Mongolia's comparatively smaller livestock size within the global context, it becomes evident that an emphasis on quality rather than quantity is imperative for producing high-quality meat and products. In our smart contract design, we incorporate a mechanism that assesses animal quality based on factors such as herd size and the pasture's carrying capacity. Recent research [23] has highlighted the significant influence of customer demand on shaping industry development, emphasizing the necessity to move beyond exclusive reliance on government policies. Despite government interventions like pasture tax [24], their effectiveness remains constrained. Our proposed solution aligns with market principles, recognizing the pivotal role of consumer demand in shaping the landscape [25].

3.1.1 NLS Dapp Smart Contract Technical Requirement

To address the challenges, we have crafted a system featuring an immutable ledger, traceability, transparency, and an inventive product grading system. This grading system assesses livestock quality by considering the ratio of herd size to pasture carrying capacity, and we will offer a comprehensive exploration of these solutions in the following section.

1. **Transparency and an immutable ledger**: Transparency and immutability requirements will be addressed by leveraging the inherent transparent and immutable nature of the blockchain within our Dapp.

2. **Traceability**: To address the traceability challenge, we will implement a solution inspired by the methods described in [6]. Each livestock in our supply chain will be equipped with a unique ear tag containing an individual identification number. When an order is placed, the livestock will be transported to a designated location, such as a slaughterhouse, for further processing. Upon arrival at the slaughterhouse, the livestock's identification number will serve as the basis for generating a new batch of QR codes. These QR codes will contain information related to the basic characteristics of the livestock as shown in Figure 3.4. They will be affixed to the packaging of each divided product, ensuring that the information is traceable throughout the supply chain.

   This approach guarantees that all the relevant information from the breeding stage to the slaughtering stage is recorded in the QR code. Subsequently, down-

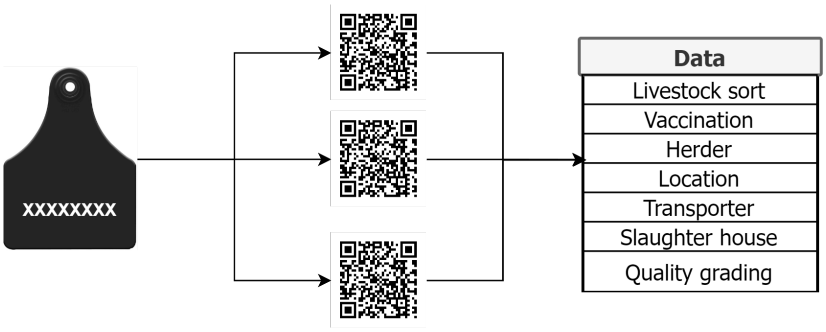

Figure 3.1: Livestock product tracking method

stream participants in the supply chain can effortlessly obtain pertinent information by simply scanning the QR code. This comprehensive traceability system ensures transparency and the ability to trace the product's journey from its origin to its final destination.

3. **Quality grading**: To assess livestock product quality and tackle overgrazing, as suggested in [26], it's essential to account for herders' irresponsible actions. This includes uncontrolled livestock expansion without considering pasture carrying capacity, which ultimately results in pasture degradation. Therefore, when calculating the grades, we will take into account the Herder-Located Pasture Capacity Ratio (PCR) and the number of livestock. We will use the livestock grazing capacity equations from the Mongolian National Statistics Committee as follows:

$$\textbf{Pasture Capacity Ratio} = \frac{\text{Number of Livestock}}{\text{Carrying Capacity of the Pasture}} \tag{3.1}$$

Where: Carrying capacity of the pasture is the maximum number of livestock that the pasture can sustainably support. It depends on factors like the size of the pasture, the type of vegetation, the quality of the forage, and the climate [27]. The number of

Livestock currently grazing on the pasture, such as cows, sheep, or any other livestock. Mongolia is divided into 21 regions called Aimag. Consequently, we will determine the pasture capacity ratio for each Aimag. A ratio of 1.0 or less indicates that the pasture is within its carrying capacity and can support the current livestock. A ratio greater than 1.0 suggests that the pasture is overgrazed [28].

Next, we will determine the appropriate number of livestock that herders can maintain based on the land size and pasture carrying capacity within each Aimag:

$$\textbf{Livestock per Herder} = \frac{\text{Carrying Capacity of the Pasture}}{\text{Total number of herders}} \tag{3.2}$$

Where: Carrying capacity of the pasture is the maximum number of livestock that the pasture can sustainably support for each Aimag land. The total number of herders is the number of herders located in that Aimag.

Next, we will calculate the herder's livestock ratio to assess whether they have exceeded the suitable livestock population according to equation 3.2.

$$\textbf{Herder Livestock Ratio} = \frac{\text{Herder's Total Livestock}}{\text{Livestock per Herder}} \tag{3.3}$$

In the final step, we will integrate the calculated Aimag pasture capacity ratio with the herder's livestock ratio to determine grades for the herder's livestock, as illustrated in Table 3.2.

For example, if the pasture ratio is 0.8, signifying a good ratio, but the herder's

Table 3.2: Livestock product quality grading point

No.	Total point	Grade
1	0-2	Grade 1
2	2.1-4	Grade 2
3	4.1-6	Grade 3
4	6.1-8	Grade 4
5	8.1-10	Grade 5
6	10-12	Grade 6

livestock ratio is 1.5, exceeding the recommended limit, the total score will be 2.3, corresponding to grade 2. This approach not only assesses livestock based on their grazing conditions but also encourages herders to prioritize quality over quantity, emphasizing responsible practices.

3.1.2 NLS Dapp Smart Contract Data Set

To ensure the accuracy and authenticity of the information within the system, as indicated in Table 3.3, the most dependable sources are the National Official Authentication system [**?**] and statistical data from the National Statistics Office [28], rendering them well-suited for practical use cases.

Table 3.3: NLS Dapp data integration source

No.	Data	Source
1	Livestock number of the Herder	National Authentication system
2	Location of the Herder	National Authentication system
3	Pasture carrying capacity of each Aimag	National Statistics Office Statistical Data
4	Total Livestock number of each Aimag	National Statistics Office Statistical Data
5	Total Herder's number each Aimag	National Statistics Office Statistical Data

However, **for the evaluation of our system, we will designate the Data as**

input during the registration process.

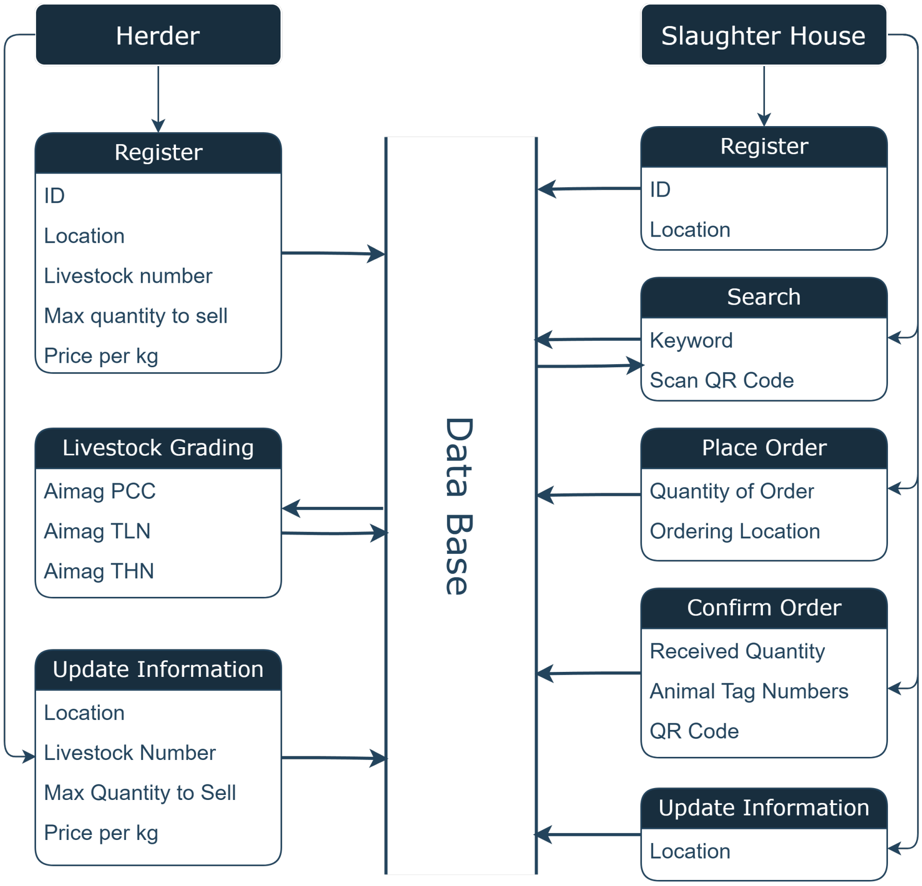

Figure 3.2: NLS Dapp smart contract data flow diagram

Figure 3.2 illustrates the data flow of the application. The initial step involves herders registering in the system, where they input their ID, corresponding Aimag name, total livestock number, the quantity of livestock available for sale, and the product price. In practical use, the application is designed to sync this information automatically with the National Authentication System. However, for evaluation purposes, we will manually input this data during registration.

After herders input their information, the application calculates the product grading based on the herder's location, pasture carrying capacity, and livestock number. The next stage involves slaughterhouses and markets. These entities input their business registration number and location when placing orders, specifying the desired quantity. Once an order is delivered, slaughterhouses input the tag numbers of each received livestock. Subsequently, they generate QR codes and tag each original livestock product.

All this information is securely stored in the database, and any application user can easily access it either by using keywords or by scanning the QR code.

3.1.3 NLS Dapp Workflow

In this section, we will outline the workflow of the NLS Decentralized Application (Dapp) within the context of the Mongolian Livestock supply chain. This new supply chain model differs from the traditional one. As shown in Figure 3.3, in this modified supply chain, all participants - herders, intermediaries, slaughterhouses, and markets - play active roles. However, a **significant transformation is evident in the role of the intermediary. In contrast to its prior influence over product prices and its pivotal role in connecting herders and markets, intermediaries now operate as transportation service providers.** This paradigm shift empowers customers to engage directly with producers, bypassing the need for intermediaries. This direct interaction enables them to collaboratively determine fair prices at each juncture of the supply chain. Consequently, this approach promotes the establishment of equitable pricing structures.

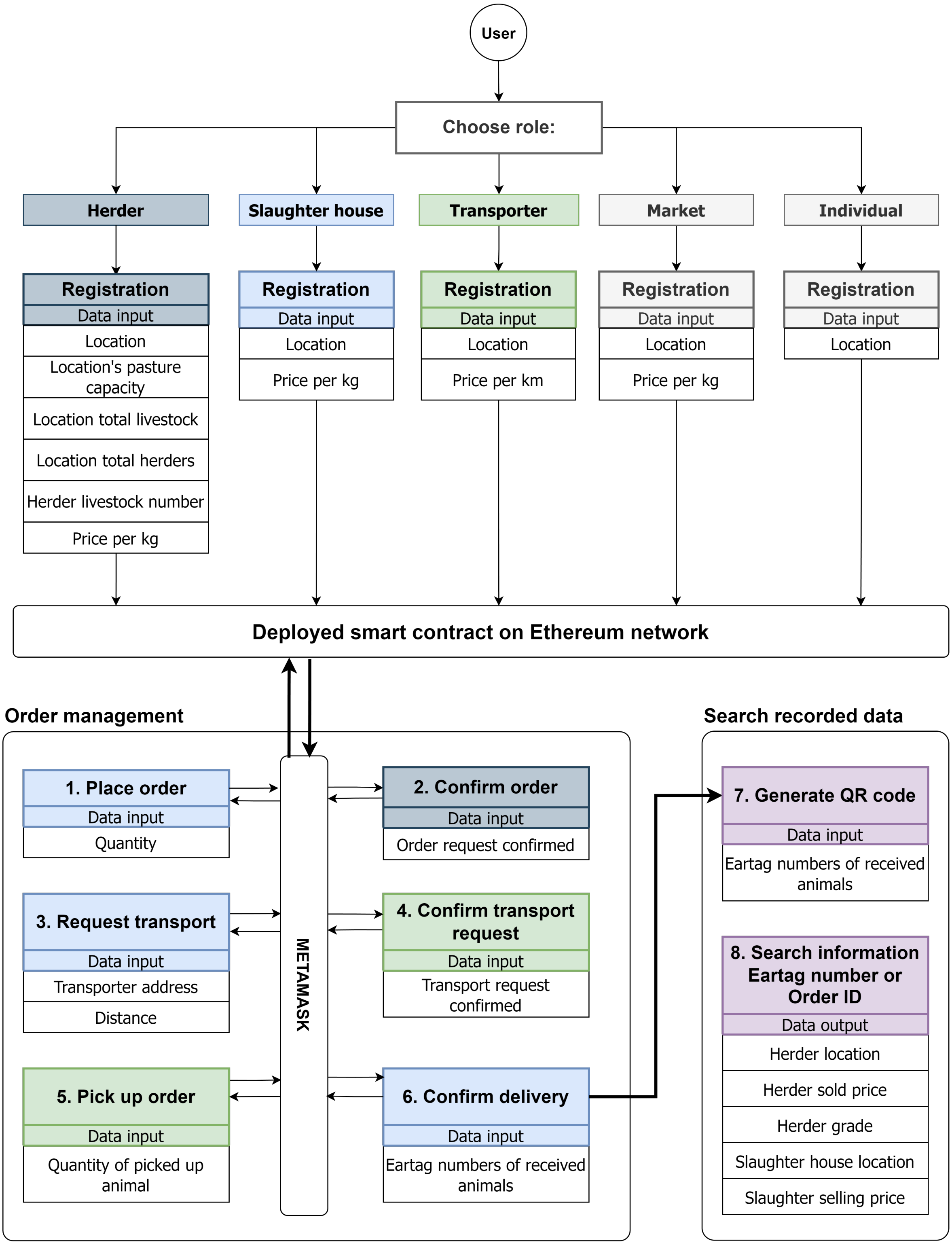

Figure 3.3: NLS Dapp Workflow Diagramm

Workflow Process:

1. Order: Herders will initiate orders directly through the application, specifying their livestock preferences. They will input details such as livestock type, quantity, location, and price. Buyers, which include slaughterhouses, wholesale markets, and individuals, will then select products based on their preferences and needs and proceed to place orders.

2. Confirm Order: Upon receiving an order, herders will carefully review the details and subsequently confirm the order's accuracy.

3. Transportation Order: After confirmation by the herder, the buyer will choose and request their preferred mode of transportation. The transportation arrangements are open to negotiation. For instance, herders may choose to include transportation costs in their product price, or the buyer can independently select and cover the transportation expenses.

4. Confirm Transportation Order: Upon receiving a delivery request, the chosen transporter will confirm the request and commence the delivery process.

5. Pick-Up Order: Upon arrival at the specified pick-up location, the transporter will input the number of livestock picked up.

6. Confirm Delivery: Upon arrival at the delivery destination, the buyer, whether a slaughterhouse, market, or individual, will input each tag number into the system and generate a QR code, which will be affixed to each processed product from its original source.

7. Subsequent Stages: In the subsequent stages of the supply chain, the order, delivery, and confirmation processes will mirror the steps outlined above.

3.1.4 NLS Dapp Smart Contract Development

Following our system requirements, we have developed a smart contract. The core logic of the smart contract is depicted in the following. The complete code of the smart contract can be found in Appendix B.

- **Role Management Functions:**
  - *chooseRole:* Allows users to select a role (Herder, Slaughterhouse, Transporter, Market) and assigns the corresponding role using AccessControl.
- **Registration Functions:**
  - *registerHerder:* Registers a herder with details like ID, location, livestock count, and pricing. Stores this information in a mapping.
  - *registerSlaughterhouse:* Registers a slaughterhouse with its ID, location, and pricing information.
  - *registerTransporter:* Registers transporters, storing their ID, truck information, pricing, and location.
- **Order Management Functions:**
  - *placeOrder:* Enables buyers to place orders with herders, specifying the herder ID and quantity. The function records the order details in a mapping.

- *confirmOrder:* Used by herders to confirm or cancel an order based on the order ID and a boolean flag.

- **Transportation and Delivery Functions:**
  - *requestTransportation:* Allows buyers to request transportation for their orders, specifying the order ID, transporter address, and distance.
  - *confirmDeliveryRequest:* Used by transporters to confirm the initiation of a delivery request.
  - *confirmPickUp:* Allows transporters to confirm the pickup of livestock, updating the order status.
  - *confirmDelivery:* Used by buyers to confirm the delivery of their order, marking it as completed.

- **Utility Functions:**
  - *calculateGradePoints:* A helper function to calculate grade points for herders based on various parameters like total livestock and carrying capacity.
  - *getGrade:* Determines the grade of a herder based on the calculated grade points.

3.2 NLS Dapp Technical Architecture

To design a technical solution that is **traceable, transparent, immutable, and programmable**, we conducted a comparative review of widely used blockchain platforms, including Ethereum, Hyperledger Fabric, and R3 Corda, as detailed in Figure

3.1. This comprehensive analysis takes into consideration factors like **ease of use, catering to the technological proficiency of herders** [29], **and scalability** [30] **to facilitate sustainable and scalable livestock management,** that integrate with other systems for legal documentation, identification, and banking. We employed a 1 to 5 scaling method to evaluate each characteristic.

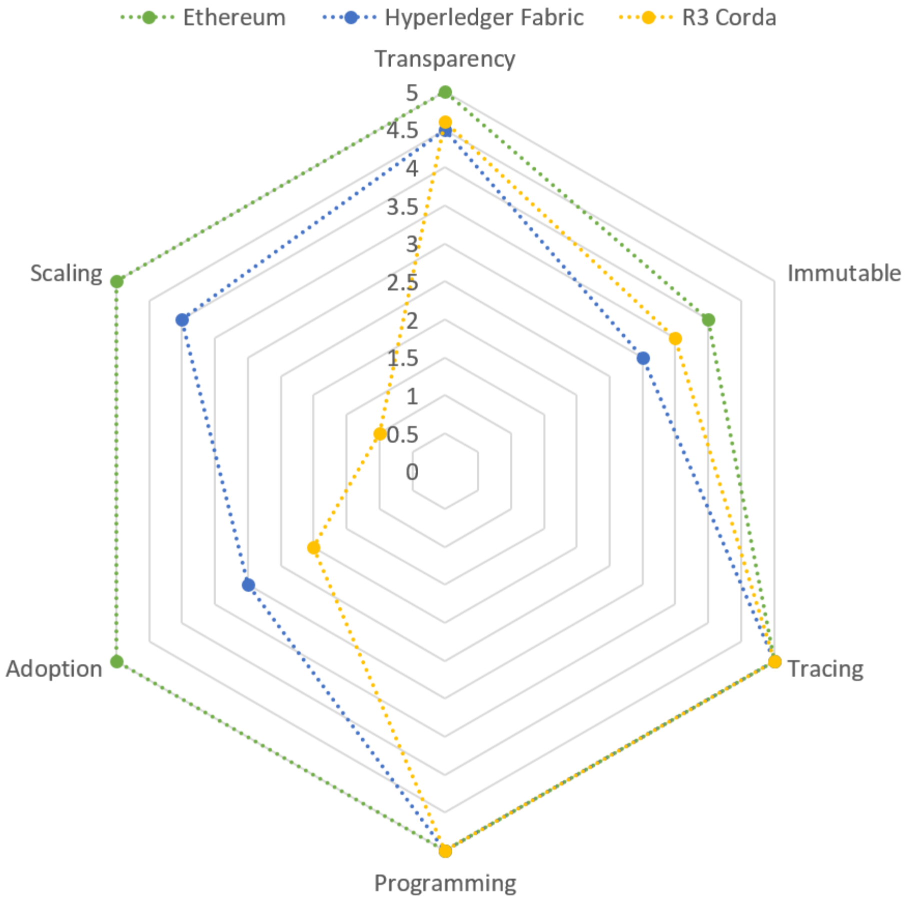

Figure 3.4: Blockchain Platforms Performance Comparison Radar Chart

Upon a thorough examination of the comparison chart, it becomes evident that all three platforms exhibit notable traits such as transparency, robust programming capabilities, and impeccable traceability, attributed to their decentralized immutable ledger feature. However, when evaluating aspects of scalability and potential for widespread adoption, Ethereum emerges as the frontrunner among the trio, primarily due to its extensive and robust ecosystem. Additionally, Ethereum demonstrates outstanding scalability, earning a commendable score of 5, indicating its capacity to handle increased workloads and adapt to evolving requirements. Consequently, we have selected Ethereum as the foundational blockchain platform for our Dapp. Consequently, our technical architecture and development environment have been meticulously crafted to seamlessly integrate with the Ethereum Dapp development ecosystem, as depicted in Figure 3.5.

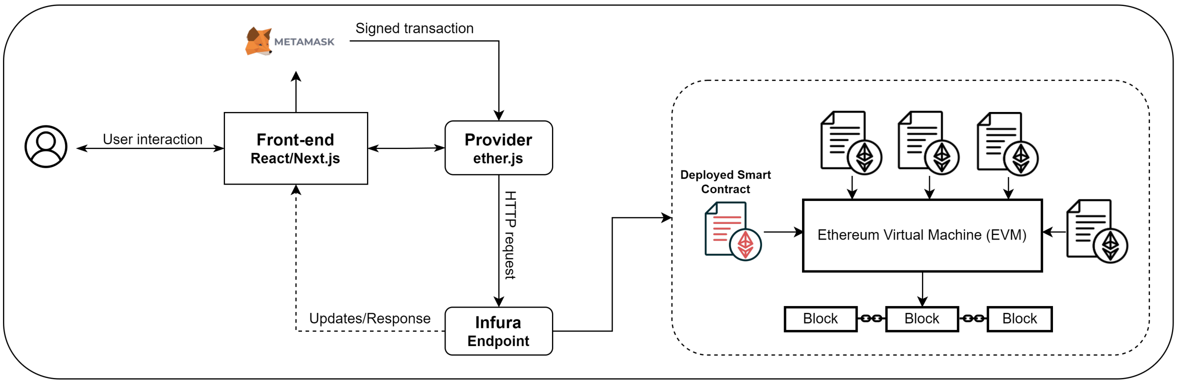

Figure 3.5: NLS Dapp Technical Architecture

As shown in the technical architecture, we have chosen Truffle as our development framework for several reasons, primarily due to its developer-friendly features. We have opted for Truffle and Ether.js, which are rapidly becoming the industry standard for smart contract development. Here's a detailed workflow of the designed technical architecture:

- User Interaction: Users interact with the DApp through the React/Next.js frontend, performing actions like sending transactions or querying data.

- Transaction Signing with MetaMask: For actions requiring transactions, the frontend uses ethers.js to create a transaction. MetaMask prompts the user to sign the transaction, ensuring security and user consent.
- Blockchain Interaction via ethers.js and Infura: Signed transactions are sent to the Ethereum blockchain using ethers.js. ethers.js communicates with the blockchain through the Infura endpoint, which provides access to the Sepolia testnet.
- Transaction Processing on Blockchain: The Ethereum blockchain on the Sepolia testnet processes the transaction.
- Feedback to User: The frontend receives updates about the transaction status (e.g., success, failure) via ethers.js and Infura. The UI is updated accordingly to inform the user.

These tools have been carefully chosen to streamline our development process and ensure a robust and efficient Dapp deployment.

Chapter 4

NLS Dapp Evaluation and Results

This chapter presents a detailed evaluation of the NLS Dapp, focusing on its functionality, performance, and cost-effectiveness. Based on this evaluation, suggestions for improvement are proposed to enhance the overall value and efficiency of the NLS Dapp.

4.1 NLS Dapp Functionality Evaluation Result

This section critically evaluates the core functionalities of the NLS Dapp, with a focus on the extent to which the application fulfills its intended requirements and objectives. The evaluation also considers the relevance of the application's features to its target user base and the overall usability of the application. For the purpose of this evaluation, five accounts were utilized to test the functionality of the prototype. The backend development was facilitated using Truffle and Ethers.js, while the front-end was developed using React and Next.js. The Dapp was tested in a local environment on the Ethereum test network, Sepolia. The subsequent paragraphs detail the results of the prototype testing, examining each function individually with its corresponding smart contract code. The complete code is available in Appendix.

1. **Connecting to MetaMask:** Users are required to connect their MetaMask wallet to interact with the Dapp.

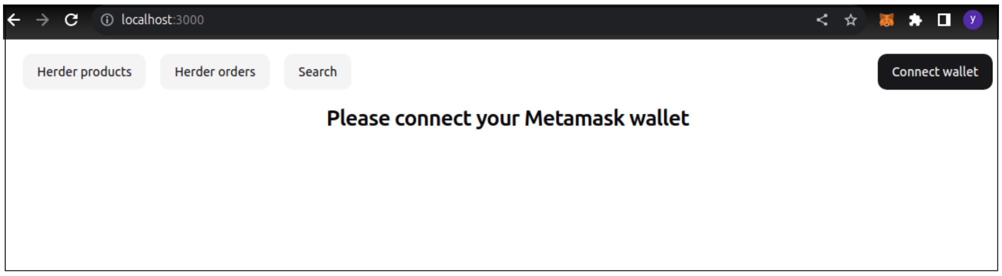

Figure 4.1: Dapp function: Connect wallet

2. **Role Selection:** Users must select one of the following roles: Herder, Slaughterhouse, Transporter, Market, or Individual, as shown in Figure 4.2.

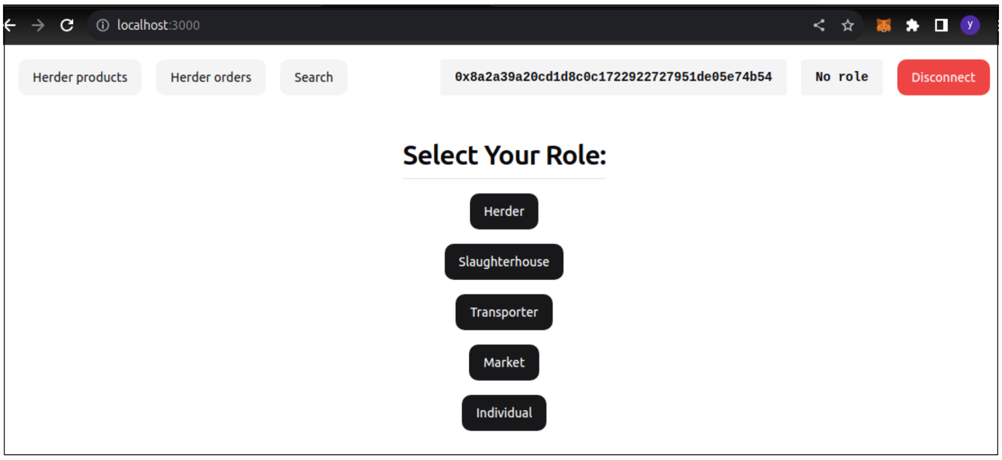

Figure 4.2: Dapp function: Choose role

3. **Registration Process:** Upon selecting a role, users are directed to the corresponding registration page. The registration inputs vary based on the chosen role. For instance, herders are required to input data such as location, total livestock number, price per kg of livestock, and other relevant details. Similarly, other roles also input role-specific data.

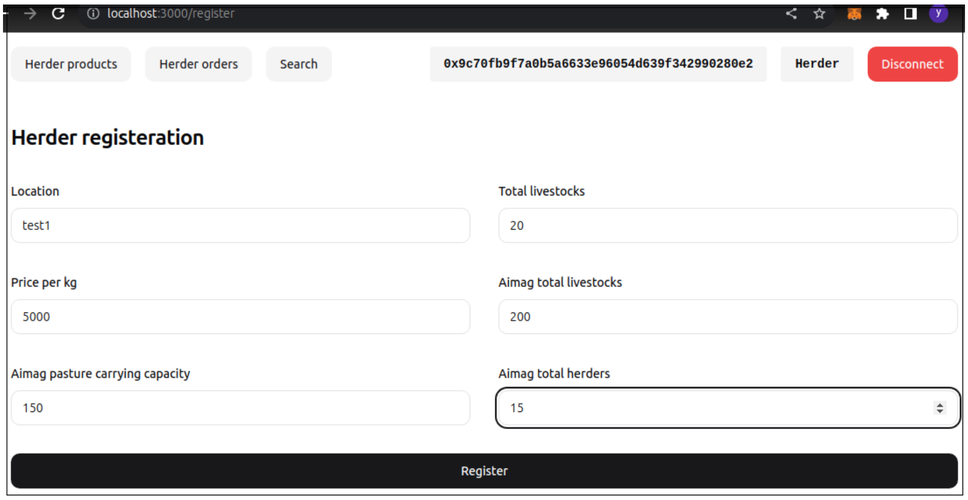

Figure 4.3: Dapp function: Register herder

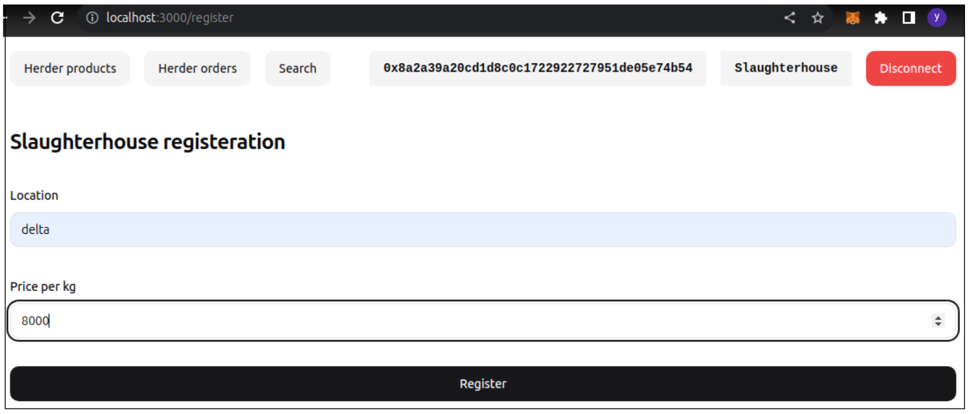

Figure 4.4: Dapp function: Register slaughterhouse

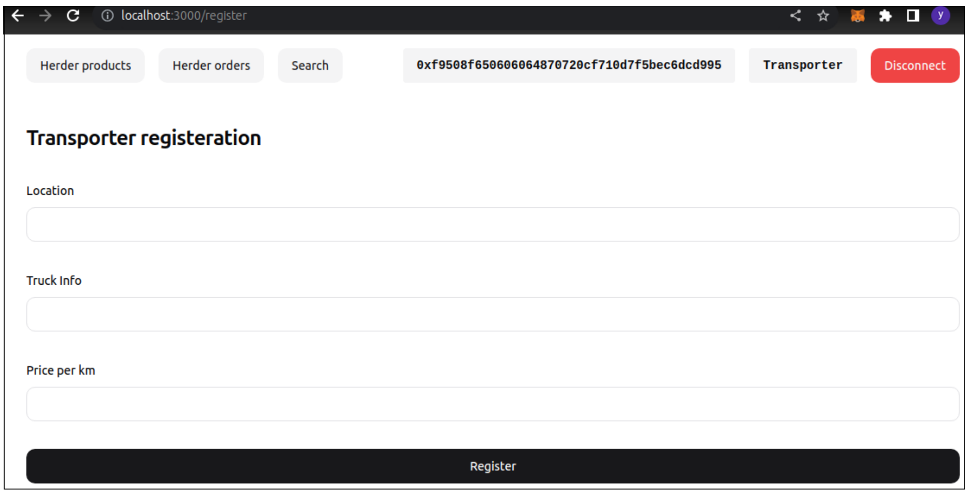

Figure 4.5: Dapp function: Register transporter

4. **Placing an order:** Once a herder's information is registered, it becomes visible on the menu. Buyers can then browse the menu to select a product, input the desired quantity, and place their order.

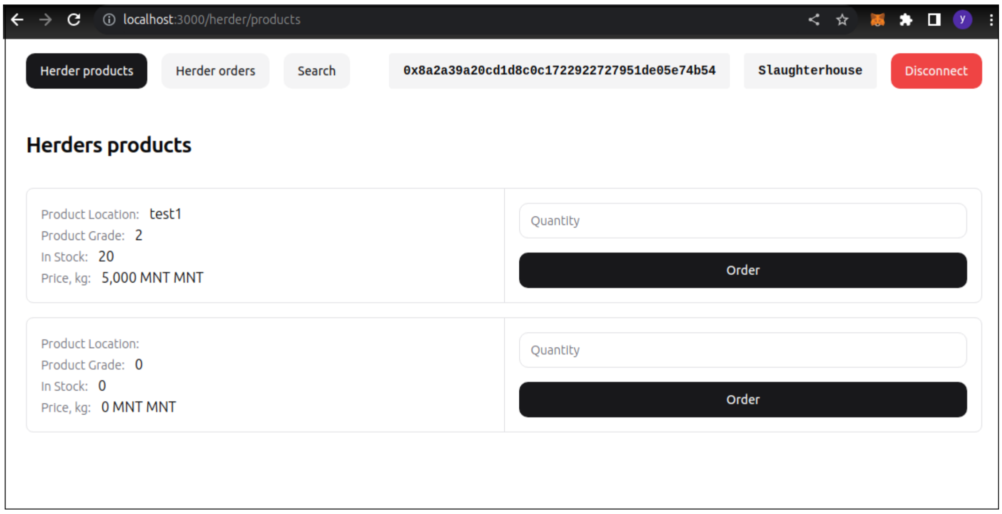

Figure 4.6: Dapp function: Placing order

5. **Requesting Transportation:** Following the seller's confirmation of the order, the buyer proceeds to request transportation for the delivery.

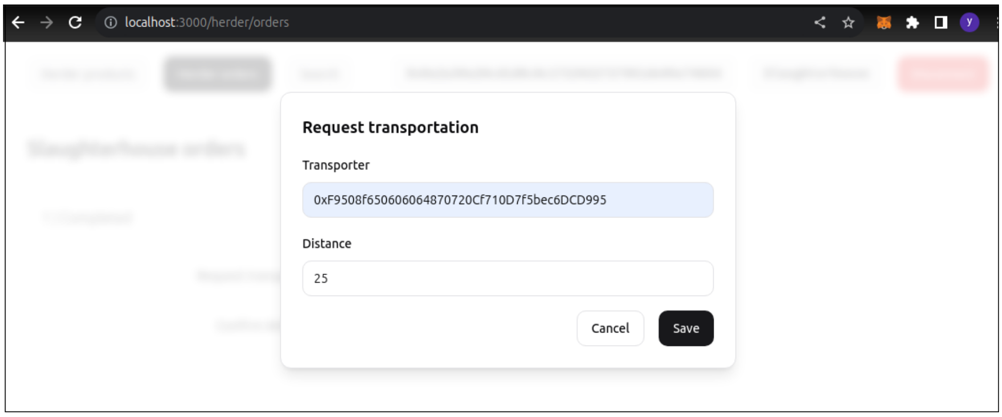

Figure 4.7: Dapp function: Transportation request

6. **Order pickup:** After the transportation request is confirmed, the transporter is responsible for going to the designated location to pick up the order. At the time

of pickup, the transporter is required to input the number of livestock collected into the Dapp, ensuring accurate tracking and record-keeping.

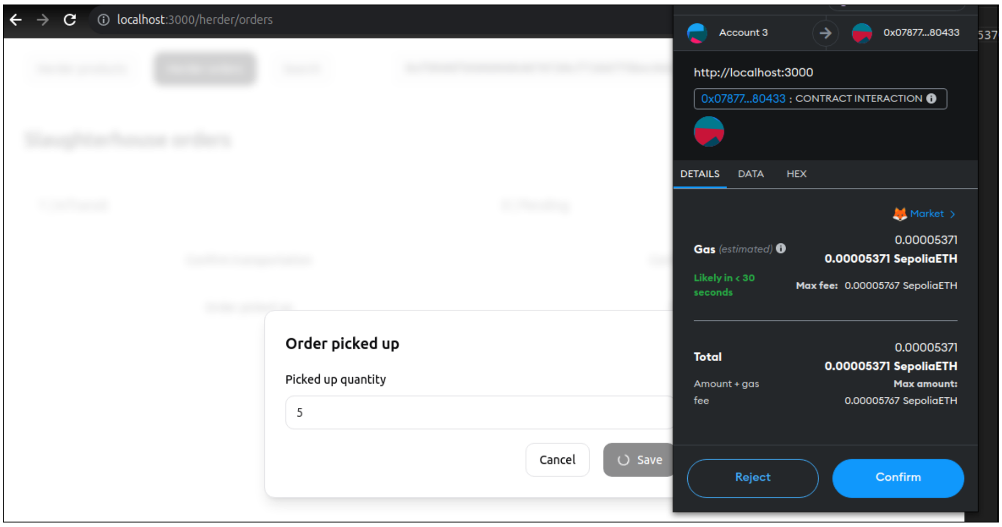

Figure 4.8: Dapp function: Order pickup

7. **Order confirmation:** Upon the delivery of the livestock, the buyer is required to input the eartag numbers for each individual animal into the Dapp. This process is crucial for tracking and verification purposes. Following the entry of these eartag numbers, the Dapp automatically generates a QR code for each number, facilitating efficient and accurate record-keeping.

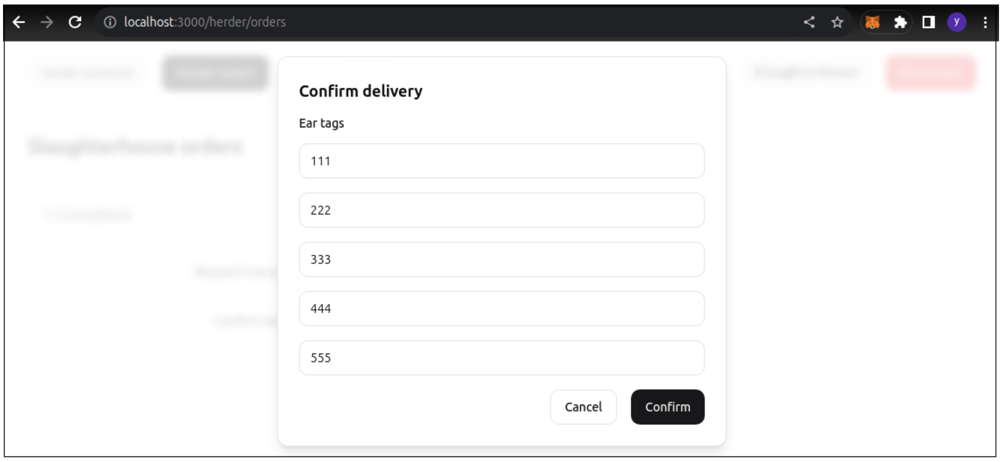

Figure 4.9: Dapp function: Order confirmation

8. **Search Information:** The Dapp is equipped with an efficient search functionality that allows users to swiftly locate product information. This can be done using either the order ID or the eartag number. The inclusion of eartag number search is particularly significant. Although our prototype was tested in a local environment where QR code scanning was not feasible, the QR codes are generated directly from these eartag numbers. Therefore, when a user scans a QR code, the Dapp retrieves and displays the same information that would appear if the eartag number were entered manually. This feature ensures that users have multiple, convenient ways to access product information.

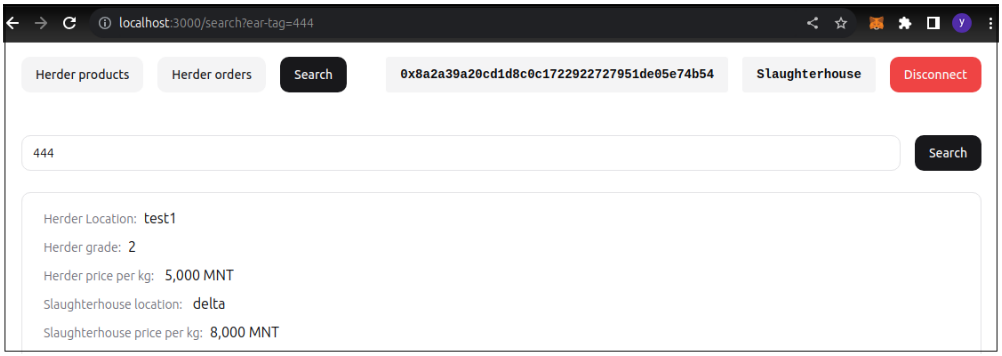

Figure 4.10: Dapp function: Order confirmation

As outlined in Chapter 3, our Dapp was designed to meet four key requirements: transparency, traceability, immutable ledger and a robust quality grading system. Upon thorough evaluation of our Dapp, we have concluded that it successfully satisfies these requirements. This conclusion is based on a detailed analysis of the Dapp's functionality and performance, demonstrating its effectiveness in providing transparency and traceability in transactions, as well as in implementing a reliable quality grading system

4.2 NLS Dapp Performance Evaluation Result

This section offers an analytical review of our smart contract's performance, focusing on three pivotal metrics: execution time, gas usage, and transaction fees. These indicators are vital for assessing the contract's efficiency and cost-effectiveness. Execution time sheds light on the responsiveness of each function, while gas usage and transaction fees underscore the costs involved in contract operations. The table below presents these metrics, collated under controlled conditions on the Sepolia network, providing a comprehensive perspective on the performance and associated costs of our smart contract's functions

Function Name	**Execution Time (ms)**	**Gas Usage**	**Transaction Fee (ETH)**	**Cost (USD)**
Contract Deployment	62 ms	5202596 gas	0.009550 ETH	21.64 USD
Role selection	62 ms	253973 gas	0.000466 ETH	1.06 USD
Registration	114 ms	587158 gas	0.001078 ETH	2.44 USD
Place order	69 ms	130686 gas	0.000240 ETH	0.54 USD
Confirm request	49 ms	78310 gas	0.000144 ETH	0.33 USD
Transport request	55 ms	83336 gas	0.000153 ETH	0.35 USD
Confirm request	65 ms	27553 gas	0.000051 ETH	0.11 USD
Order pick-up	62 ms	28252 gas	0.000052 ETH	0.12 USD
Confirm order	102 ms	77329 gas	0.000142 ETH	0.32 USD

Table 4.1: Performance metrics of smart contract functions

Our analysis of the NLS DApp underscores two primary findings: the execution times are reasonable but could be affected by Ethereum's network congestion, and the transaction costs pose significant implications for end-users. While execution times are generally satisfactory, scalability solutions could mitigate the impact of network congestion. The high transaction costs on the Ethereum platform raise concerns about the DApp's commercial viability, necessitating exploration of cost-reduction strategies such as optimizing smart contract efficiency, exploring alternative blockchain platforms, or modifying the fee structure. The subsequent sections will propose technical and economic strategies to enhance the DApp's performance and affordability, aiming to boost its practicality and appeal in a commercial setting.

4.3 NLS Dapp Improvement Suggestions

The evaluation results provide valuable insights into the strengths and areas for improvement within the DApp's current framework. Based on these findings, we propose the following targeted improvement suggestions to address the specific challenges identified:

Authentic Data Source Integration

- Integration with national databases and statistical data sources will be pursued, as emphasized in Chapter 3, to bolster data reliability and authenticity. This includes synchronizing updated user and seller information, as well as statistical data such as pasture conditions, to enhance data accuracy and stakeholder confidence.

Payment System Reform and Transaction Fee Strategy

- *Banking App Integration*: To facilitate user familiarity, payments will be integrated through common banking applications, providing a conventional and trusted transaction medium.

- *Transactional Cost Absorption*: The entity behind the DApp will absorb transaction fees. This approach includes smart contract reconfiguration, fee reserve management, establishing alternative revenue channels, ensuring continuous monitoring for sustainability, maintaining transparent communication with users, and adhering to financial and cryptocurrency regulations.

Replacing physical Eartags

- In response to the limitations of physical Ear tags, which can be lost or become unreadable due to harsh environmental conditions, an alternative approach could be the use of unique ID numbers for each livestock. This method would involve assigning and maintaining a digital database of distinct IDs for each animal, circumventing the challenges posed by physical tags. The digital IDs would be less susceptible to environmental wear and tear and would not get lost, ensuring consistent and reliable identification throughout the livestock's lifecycle. This approach streamlines the tracking process and enhances the accuracy and reliability of livestock management systems.

Storage Capacity Enhancement

- Adoption of the InterPlanetary File System (IPFS) is planned for decentralized and efficient data storage. This approach aims to manage the increasing data volume and user base, thereby enhancing system performance and data management.

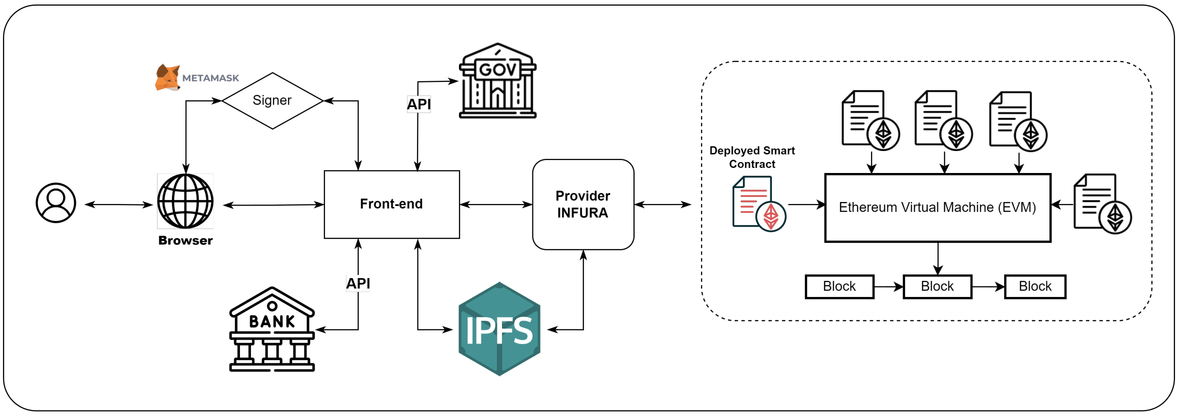

Figure 4.11: Dapp advanced technical architecture

4.4 Conclusion

In conclusion, these targeted enhancements aim to significantly improve the NLS DApp in terms of user accessibility, data integrity, financial feasibility, and storage efficiency. This strategic approach is pivotal for enhancing the DApp's usability, reliability, and commercial viability, thereby setting a strong foundation for its continued development and success in the dynamic blockchain environment.

Summary

Decentralized Herding: Improving Semi-Nomadic Livestock Supply Chain through Blockchain

The thesis "Decentralized Herding: Improving Semi-Nomadic Livestock Supply Chain through Blockchain" presents a comprehensive analysis and solution for enhancing semi-nomadic livestock supply chains, using Mongolia as a primary case study. The initial chapters lay the foundation by exploring blockchain technology's principles and potential applications in agriculture, emphasizing the need for improved traceability, data integrity, and supply chain management.

A significant portion of the thesis is dedicated to the design, development, and technical detailing of the NLS DApp, a blockchain-based application tailored for the livestock supply chain. This includes the creation of smart contracts, the architecture of the DApp, and the integration with blockchain platforms, focusing on user-friendliness and scalability.

The thesis then progresses to a detailed evaluation of the NLS DApp. It assesses the application's functionality and performance, examining key metrics such as execution time, gas usage, and transaction costs. This evaluation leads to a set of proposed improvements to enhance the DApp's user interface, data source integration, payment system, and storage solutions.

In its concluding chapters, the thesis consolidates these findings and suggestions,

emphasizing the significance of the enhancements for optimizing the DApp's functionality, user experience, and commercial viability. The study presents a strategic approach to utilizing blockchain in improving semi-nomadic livestock supply chains, offering a model that could be adapted and applied to similar supply chains globally.

References

1. V. Bahga, Arshdeep & Madisetti, "Blockchain platform for industrial internet of things," *Journal of Software Engineering and Applications*, vol. 9, no. 910036, pp. 533–546, 2016.

2. V. Clincy and H. Shahriar, "Blockchain development platform comparison," *IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 43, 2019.

3. L. Ismail and H. Materwala, "A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions," *Symmetry*, vol. 11, no. 10, 2019.

4. J. Golosova and A. Romanovs, "The advantages and disadvantages of the blockchain technology," *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, pp. 1–6, 2018.

5. V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," technical report, Ethereum foundation, 2014.

6. G. Aiyun, D. Debin, and H. Zhongwei, "Rfid in the livestock supply chain management: Application and development," *2010 International Conference on E-Business and E-Government*, pp. 3424–3427, 2010.

7. N. S. O. of Mongolia, "Livestock industry annual report 2022," Statistical Report MCE-BR-2022-04, NSA, Mongolia, U.B., April 2022. Annual statistical report.

8. A. Gantulga, "Survey on the domestic and export meat value chain in mongolia," survey report, United Nations Conference on Trade and Development (UNCTAD), 2021.

9. S. Machiraju, "Livestock commercialization project in mongolia," project report, The World Bank, 2019.

10. I. M. Fund, "Greening growth in mongolia," 2019. [Accessed: 09 14, 2023].

11. H. D. Jie Han and Z. Gu, "Sandstorms and desertification in mongolia, an example of future climate events: a review," *Environmental Chemistry Letters*, vol. 19, p. 4063–4073, 2021.

12. M. Sharma, A. A. Khalil, and T. Daim, "Blockchain technology adoption: Multinational analysis of the agriculture supply chain," *IEEE Transactions on Engineering Management*, pp. 1–18, 2022.

13. A. Asfarian, K. I. Hilmi, and I. Hermadi, "Preliminary user studies on consumer perception towards blockchain-based livestock traceability platform in indonesia: An implication to design," *2020 International Conference on Computer Science and Its Application in Agriculture (ICOSICA)*, pp. 1–6, 2020.

14. U. S. Kushankur Dey, "Blockchain for sustainable e-agriculture: Literature review, architecture for data management, and implications," *Journal of Cleaner Production*, vol. 316, 2021.

15. W. A. C. L. Rathnayka and K. P. N. Jayasena, "Permissioned blockchain plat-

form to enhance scalability, security and performance issues in livestock farms in sri lanka,," *2021 3rd International Conference on Advancements in Computing (ICAC)*, vol. 3, pp. 461–466, 2021.

16. Q. Zheng, N. Lin, D. Fu, T. Liu, and Zhu, "Smart contract-based agricultural service platform for drone plant protection operation optimization," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

17. I. Afrianto, T. Djatna, Y. Arkeman, I. Sukaesih Sitanggang, and I. Hermadi, "Disrupting agro-industry supply chain in indonesia with blockchain technology: Current and future challenges," *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1–6, 2020.

18. W. P. T. M. M. L. Shoufeng Cao, Marcus Foth, "A blockchain-based multisignature approach for supply chain governance: A use case from the australian beef industry," *Blockchain: Research and Applications*, vol. 3, no. 4, pp. 1–6, 2022.

19. Z. X. W. X. e. a. Peng, X., "Construction of rice supply chain supervision model driven by blockchain smart contract," *Scientific Reports*, vol. 12, no. 20984, 2022.

20. T. A. A. Kampan, K. Tsusaka, "Adoption of blockchain technology for enhanced traceability of livestock-based products," *Sustainability*, vol. 14, no. 20, p. 13148, 2022.

21. S. Mondal, K. P. Wijewardena, S. Karuppuswami, N. Kriti, D. Kumar, and P. Chahal, "Blockchain inspired rfid-based information architecture for food supply chain," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5803–5813, 2019.

22. F.-G. R. Jamsranjav, C. and B. Adya, “Opportunities to integrate herders’ indicators into formal rangeland monitoring: an example from mongolia,” *Ecological Applications*, vol. 29, 2019.

23. P. P.-L. . E. G.-M. Marta Arenas-Jal, J. M. Suñé-Negre, “Trends in the food and sports nutrition industry: A review, critical reviews in food science and nutrition,” *Critical Reviews in Food Science and Nutrition*, vol. 60, no. 14, pp. 2405–2421, 2019.

24. “Pasture tax of mongolia: Official tax on number of livestock,” 2020. https://legalinfo.mn/mn/detail/15729.

25. N. R. A. Ajay Kumar, Ravi Shankar, “A big data driven framework for demand-driven forecasting with effects of marketing-mix variables,” *Industrial Marketing Management*, vol. 90, pp. 493–507, 2020.

26. T. C. for Policy Research (CPR), “Baseline study report: Feasibility study on pasture use fee,” statistical report, The Center for Policy Research (CPR), https://www.undp.org/sites/g/files/zskgke326/files/migration/mn, 2017.

27. ”Yu, H., Liu, Bt., Wang, Gx. et al”, “Grass-livestock balance based grassland ecological carrying capability and sustainable strategy in the yellow river source national park, tibet plateau, china,” *Journal of Mountain Science*, vol. 18, p. 2201–2211, 2021.

28. M. of Food and Agriculture, “National report on the rangeland health of mon-

golia," statistical report, Swiss Agency for Development and Cooperation SDC, https://jornada.nmsu.edu/files/Mongolia-Rangeland-health-Report$_E N.pdf$, 2015.

29. X. R. L. J. B. Jiabao Lin, Lei Li, "How do agribusinesses thrive through complexity? the pivotal role of e-commerce capability and business agility," *Decision Support Systems*, vol. 135, no. 113342, 2020.

30. Y. L. T. Julien Polge, Jérémy Robert, "Permissioned blockchain frameworks in the industry: A comparison," *ICT Express*, vol. 7, no. 2, pp. 229–233, 2021.

Appendix A

Dapp development configuration

Network configuration:

Network name	Sepolia
Network ID	11155111
Infura API key	sepolia.infura.io/v3/59ee3c35e9254901a44338f550fa2139

Table A.1: Smart contract deployment configuration

Transaction log for function evaluation process:

Function	**Block Explorer Address**
Contract deployment	https://sepolia.etherscan.io/tx/0x34e623f864c296eba7b1aaa21c38486223c18cb55f31ab87cce77aaed5a1fbcf
Choose role function	https://sepolia.etherscan.io/tx/0xfbd7224a46eee6a02264c1b2d318c5c2109e693e120a71132cb18a9dda06f5e1
Register herder function	https://sepolia.etherscan.io/tx/0x402825a4005aa31ab9189d95f2a5ac2572a80d048cf382b40be65f83f9b8cf7d
Place order function	https://sepolia.etherscan.io/tx/0xb76a821790f1ecc25652c61151a1bb519e1a0ee5704fcf2c6eaf055aa294dc49
Request transportation function	https://sepolia.etherscan.io/tx/0x0cd406c50396cd246391d90874277e05e17bf6a49a313dd9d1fe9687725d54fa
Pick up order function	https://sepolia.etherscan.io/tx/0xda56cdc422746fb2abcc717d869038c8ef19f6c44a924c01d4a55f97e0be0252
Confirm delivery function	https://sepolia.etherscan.io/tx/0xaaf8506e62bdd43087fc18904f0e59b691d27200fd4c6ae643d5dd5450bb043b

Table A.2: Transaction log for function evaluation process

Appendix B

Smart Contract Code

```
// SPDX-License-Identifier: MIT
// This code is written by Yesuilen.G, Infonet lab, GIST
pragma solidity ^0.8.22;
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/access/AccessControl.sol";
contract Supplychain is Ownable, AccessControl {
    // Define roles with the AccessControl framework
    bytes32 public constant HERDER_ROLE = keccak256("HERDER");
    bytes32 public constant SLAUGHTERHOUSE_ROLE = keccak256("SLAUGHTERHOUSE");
    bytes32 public constant TRANSPORTER_ROLE = keccak256("TRANSPORTER");
    bytes32 public constant MARKET_ROLE = keccak256("MARKET");
    //bytes32 public constant INDIVIDUAL_ROLE = keccak256("INDIVIDUAL");
    // Define enums with consistent naming
    enum OrderStatus { PENDING, CONFIRMED, IN_TRANSIT, COMPLETED, CANCELED }
    enum RoleChoice { Herder, Slaughterhouse, Transporter, Market, Individual }
    // Counter for generating unique IDs
```

```
    uint256 private herderIdCounter = 1;
    uint256 private slaughterhouseIdCounter = 1;
    uint256 private transporterIdCounter = 1;
    // Define structs for each stakeholder
    struct Herder {
        uint256 id;
        string location;
        uint256 grade;
        uint256 totalLivestock; // Total number of livestock
        uint256 pricePerKg; // Price per kilogram
        bool registered;
        uint256 aimagTotalLivestock;
        uint256 aimagPastureCarryingCapacity;
        uint256 aimagTotalHerderNumber;
    }
    struct Slaughterhouse {
        uint256 id;
        string location;
        uint256 pricePerKg;
        bool registered;
    }
    struct Transporter {
        uint256 id;
        string truckInfo;
        uint256 pricePerKm;
        bool registered;
```

```
        string location;
    }
    // Define struct for orders
    struct Order {
        uint256 id;
        address buyer;
        address seller;
        uint256 quantity;
        uint256 price;
        uint256 distance; // Distance in kilometers or any unit of measure
        uint256 transportationCost; // Calculated based on distance and pricePerKm
        OrderStatus status;
        uint256[] earTagNumber;
    }
    // Define state variables
    mapping(uint256 => Herder) public herders;
    mapping(uint256 => Slaughterhouse) public slaughterhouses;
    mapping(uint256 => Transporter) public transporters;
    mapping(uint256 => Order) public orders;
    // Mapping to link an Ethereum address with its numerical herder ID
    mapping(address => uint256) public herderAddressToId;
    mapping(address => uint256) public slaughterhouseAddressToId;
    mapping(address => uint256) public transporterAddressToId;
```

```
    mapping(uint256 => uint256) public earTagToOrderId;
    mapping(address => bool) public hasChosenRole;
    mapping(address => RoleChoice) private userRoles;
    uint256 public nextOrderId = 1;
    mapping(uint256 => address) public herderIdToAddress;
    mapping(uint256 => address) public slaughterhouseIdToAddress;
    mapping(uint256 => address) public transporterIdToAddress;
    // Function for users to choose a role
    function chooseRole(RoleChoice choice) public {
        require(!hasChosenRole[msg.sender], "User has already chosen a role");
        bytes32 role;
        if (choice == RoleChoice.Herder) {
            role = HERDER_ROLE;
        } else if (choice == RoleChoice.Slaughterhouse) {
            role = SLAUGHTERHOUSE_ROLE;
        } else if (choice == RoleChoice.Transporter) {
            role = TRANSPORTER_ROLE;
        } else {
            revert("Invalid role choice");
        }
        _grantRole(role, msg.sender);
        userRoles[msg.sender] = choice; // Store the user's chosen role
        hasChosenRole[msg.sender] = true;
    }
```

```
    // Function to register a herder
    function registerHerder(
        string memory location,
        uint256 totalLivestock,
        uint256 pricePerKg,
        uint256 aimagTotalLivestock,
        uint256 aimagPastureCarryingCapacity,
        uint256 aimagTotalHerderNumber
    ) public {
        require(hasRole(HERDER_ROLE, msg.sender), "Caller is not a herder");
        require(herderAddressToId[msg.sender] == 0, "Herder already registered.");
        uint256 herderId = herderIdCounter++;
        herderAddressToId[msg.sender] = herderId;
        herderIdToAddress[herderId] = msg.sender;
        uint256 gradePoints = calculateGradePoints(aimagTotalLivestock, aimagPastureCarryingCapacity, totalLivestock, aimagTotalHerderNumber);
        uint256 grade = getGrade(gradePoints);
        Herder storage newHerder = herders[herderId];
        newHerder.id = herderId;
        newHerder.totalLivestock = totalLivestock;
        newHerder.pricePerKg = pricePerKg;
        newHerder.location = location;
        newHerder.grade = grade;
```

```
        newHerder.registered = true;
        newHerder.aimagTotalLivestock = aimagTotalLivestock;
        newHerder.aimagPastureCarryingCapacity = aimagPastureCarryingCapacity;
        newHerder.aimagTotalHerderNumber = aimagTotalHerderNumber;
        herderAddressToId[msg.sender] = herderId;
        emit RegisteredHerder(herderId, grade, msg.sender);
    }
    function calculateGradePoints(
        uint256 aimagTotalLivestock,
        uint256 aimagPastureCarryingCapacity,
        uint256 totalLivestock,
        uint256 aimagTotalHerderNumber
        ) internal pure returns (uint256) {
        uint256 pcr = aimagTotalLivestock  / aimagPastureCarryingCapacity; // Multiply by 10 to preserve some decimal accuracy
        uint256 livestockPerHerder = aimagPastureCarryingCapacity  / aimagTotalHerderNumber;
        uint256 gradePoints = pcr + (totalLivestock  / livestockPerHerder) ; // Adjust the final result
        return gradePoints;
        }
    function getGrade(uint256 gradePoints) internal pure returns (uint256) {
        if (gradePoints >= 0 && gradePoints <= 2) return 1;
```

```
    if (gradePoints > 2 && gradePoints <= 4) return 2;
    if (gradePoints > 4 && gradePoints <= 6) return 3;
    if (gradePoints > 6 && gradePoints <= 8) return 4;
    if (gradePoints > 8 && gradePoints <= 10) return 5;
    return 6; // For gradePoints greater than 10
    }
    // Function to register a slaughterhouse
    function registerSlaughterhouse(string memory location, uint256
        pricePerKg) public {
        require(hasRole(SLAUGHTERHOUSE_ROLE, msg.sender), "Caller is
            not a slaughterhouse");
        require(slaughterhouseAddressToId[msg.sender] == 0, "
            Slaughterhouse already registered.");
        uint256 id = slaughterhouseIdCounter++;
        Slaughterhouse storage newSlaughterhouse = slaughterhouses[id
            ];
        newSlaughterhouse.id = id;
        newSlaughterhouse.location = location;
        newSlaughterhouse.pricePerKg = pricePerKg;
        newSlaughterhouse.registered = true;
        slaughterhouseAddressToId[msg.sender] = id;
    }
    // Function to register a transporter
    function registerTransporter(string memory location, string memory
         truckInfo, uint256 pricePerKm) public {
        require(hasRole(TRANSPORTER_ROLE, msg.sender), "Caller is not
```

```
a transporter");
        require(transporterAddressToId[msg.sender] == 0, "Transporter already registered.");
        uint256 id = transporterIdCounter++;
        Transporter storage newTransporter = transporters[id];
        newTransporter.id = id;
        newTransporter.truckInfo = truckInfo;
        newTransporter.pricePerKm = pricePerKm;
        newTransporter.location = location;
        newTransporter.registered = true;
        transporterAddressToId[msg.sender] = id;
    }
    // Function to place an order
function placeOrder(uint256 herderId, uint256 quantity) public {
    require(herderId != 0 && herderId < herderIdCounter, "Invalid herder ID.");
    require(herderAddressToId[msg.sender] != herderId, "Cannot place an order with yourself.");
    Herder storage herder = herders[herderId];
    require(herder.registered, "Seller is not a registered herder.");
    require(herder.totalLivestock >= quantity, "Not enough livestock available.");
    uint256 price = herder.pricePerKg * quantity;
    Order storage newOrder = orders[nextOrderId];
    newOrder.id = nextOrderId;
    newOrder.buyer = msg.sender;
```

```
        newOrder.seller = herderIdToAddress[herderId]; // Use the herder's address associated with the ID
        newOrder.quantity = quantity;
        newOrder.price = price;
        newOrder.status = OrderStatus.PENDING;
        newOrder.earTagNumber = new uint256[](0);
        emit OrderPlaced(newOrder.id, msg.sender, price, quantity);
    }
        // Function to confirm an order
        function confirmOrder(uint256 orderId, bool confirm) public onlyHerder {
            require(orders[orderId].seller == msg.sender, "Only the seller can confirm the order");
            require(orders[orderId].status == OrderStatus.PENDING, "Order is not pending.");
            if (confirm) {
                orders[orderId].status = OrderStatus.CONFIRMED;
                emit OrderConfirmed(orderId);
            } else {
                orders[orderId].status = OrderStatus.CANCELED;
                emit OrderCanceled(orderId);
            }
        }
        // Function to request transportation
        function requestTransportation(uint256 orderId, address transporter, uint256 distance) public {
```

```
        require(orders[orderId].buyer == msg.sender, "Only the buyer can request transportation");
        require(orders[orderId].status == OrderStatus.CONFIRMED, "Order not confirmed yet");
        // Calculate transportation cost
        uint256 transporterId = transporterAddressToId[transporter];
        Transporter storage transporterStruct = transporters[transporterId];
        require(transporterStruct.registered, "Transporter not registered");
        uint256 transportationCost = distance * transporterStruct.pricePerKm;
        // Update the order with transportation details
        Order storage order = orders[orderId];
        order.distance = distance;
        order.transportationCost = transportationCost;
        order.status = OrderStatus.IN_TRANSIT;
        emit TransportationRequested(orderId, transporter, distance);
    }
    // Function to confirm delivery request
    function confirmDeliveryRequest(uint256 orderId) public onlyTransporter {
        require(orders[orderId].status == OrderStatus.IN_TRANSIT, "Transport request not in transit state");
        emit DeliveryConfirmed(orderId);
    }
```

```
    // Function to confirm pick up
    function confirmPickUp(uint256 orderId, uint256 quantityPickedUp) public onlyTransporter {
        require(orders[orderId].status == OrderStatus.IN_TRANSIT, "Order must be in transit");
        emit LivestockPickedUp(orderId, quantityPickedUp);
    }
    // Function to confirm delivery
    function confirmDelivery(uint256 orderId, uint256[] calldata earTagNumbers) public {
        require(orders[orderId].status == OrderStatus.IN_TRANSIT, "Order must be in transit");
        require(orders[orderId].buyer == msg.sender, "Only the buyer can confirm delivery");
        for (uint i = 0; i < earTagNumbers.length; i++) {
            earTagToOrderId[earTagNumbers[i]] = orderId;
        }
        Order storage order = orders[orderId]; // Fix here, properly access the order object
        order.status = OrderStatus.COMPLETED;
        emit LivestockDelivered(orderId, earTagNumbers);
    }
    function getLivestockDataByEarTag(uint256 earTagNumber) public view returns (
    string memory herderLocation,
    uint256 herderGrade,
```

```
    uint256 herderPricePerKg,
    string memory slaughterhouseLocation,
    uint256 slaughterhousePricePerKg,
    string memory transporterInfo,
    uint256 transporterPricePerKm
) {
    uint256 orderId = earTagToOrderId[earTagNumber];
    Order memory order = orders[orderId];
    Herder memory herder = herders[herderAddressToId[order.seller]];
    uint256 slaughterhouseId = slaughterhouseAddressToId[order.buyer];
    Slaughterhouse memory slaughterhouse = slaughterhouses[
        slaughterhouseId];
    uint256 transporterId = transporterAddressToId[order.buyer];
    Transporter memory transporter = transporters[transporterId];
    // Return the requested details
    return (
        herder.location,
        herder.grade,
        herder.pricePerKg,
        slaughterhouse.location,
        slaughterhouse.pricePerKg,
        transporter.truckInfo,
        transporter.pricePerKm
    );
}
}
```