

A GENERIC, SCALABLE AND SECURE DATA DRIVEN SUPPLY CHAIN
CONNECTIVITY FRAMEWORK FOR ENABLING COLLABORATION, KNOWLEDGE
TRANSFER AND TRACEABILITY

By

Salman Ali

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science—Doctor of Philosophy

2024

ABSTRACT

The food supply chain network is a complex system involving various subsystems such as stock management, feed harvesting, cold storage, transportation, retail businesses, and regulatory certifications for food production. Throughout the food supply chain, major subsystems are owned by private organizations that share either little, or no information with other organizations. The restricted information flow, due to the fragmented and disjoint nature of the supply chain, results in reduced trust and traceability. With no knowledge shared, from the sequence of processes at different stages of the supply chain, an opportunity is lost to optimize the chain for better economic and environmental outcomes. The current technology in place for data communication, which relies on private and centralized ledgers, does not facilitate in the dissemination of critical information across the food supply chain network. This form of technology further limits the ability to collaborate on traceability, knowledge transfer and federated machine learning applications, because different subsets of the common data are owned by different private entities.

In this thesis, we propose a decentralized and distributed supply chain connectivity and collaboration framework that is paired with blockchain technology, distributed resources, application and methods to enable reliable and non-pervasive food supply chain pertinent data consumption, data management, information extraction and knowledge transfer in a collaborative way. The proposed framework facilitates timely dissemination of critical information that is common to collaborating organizations, without any concerns for privacy, security and loss of data control. As a result of the helpful information dissemination from end-to-end, trust, transparency, traceability and collaboration are promoted.

The technical contribution of this thesis lies in the generic, scalable, decentralized and distributed user controlled framework, that allows extracting and utilizing vital information from organizational data at different levels of the supply chain, along with its dissemination from end-to-end without any concerns of privacy, security, immutability and loss of data ownership. Seamless configuration and integration of distributed application and resources to support and enable, reliable federated machine learning data pipelines, makes the framework ideal for collaboration

among distributed and disjoint organizations in the food supply chain network. Taking into account complex supply chains, the proposed extensible connectivity and collaboration framework allows integrating major types of information sources, (for example streaming data, hybrid databases, data feeds and static data sinks), while ensuring reliable and tamper-proof traceability, as data flows through collaboration channels. Information in the proposed framework is extracted and securely propagated using an integrated hierarchical blockchain infrastructure, coupled with distributed data storage, that is configured in a private network setting according to the organizational layout of participating supply chain actors.

The organization controlled communication channels that are enabled in ad-hoc scenarios for collaboration, facilitate participants to communicate policies and decisions along with the implementation of numerous useful supply chain applications. Examples of some food supply chain related policies and applications that can be implemented by our proposed framework include, trading of carbon credits, tracking cattle in beef supply chain, jointly managing greenhouse gas emissions and optimizing end-to-end supply chain resource consumption. Strategies and techniques for protecting and securing the proposed distributed blockchain-based framework from the viewpoint of user accessibility, data integrity, confidentiality and privacy are also incorporated. The proposed framework takes into account, considerations for detailed software application level security measures to further enhance user trust. By incorporating a ‘Beef Supply Chain’ example scenario, evaluation of an implemented application (named BeefMesh) has been done to prove its efficacy for collaboration, policy sharing, traceability, secure federated learning architectures, knowledge transfer and increased value for supply chain participants.

Copyright by
SALMAN ALI
2024

ACKNOWLEDGEMENTS

To begin, I extend my heartfelt appreciation to my advisor, Dr. Wolfgang Banzhaf. His welcoming demeanor, unwavering patience, and steadfast support has been my anchor throughout this journey. His enthusiasm for knowledge and science, combined with his collaborative spirit and insightful guidance, have significantly influenced my thinking and were crucial in bringing this manuscript to reality.

I also want to acknowledge my committee members, Dr. Cedric Gondro, Dr. Qiben Yan, and Dr. Charles Owen. The constructive feedback, kindness, and guidance that I received from my committee members were instrumental in steering me toward the right research topic. I would particularly like to express my deepest gratitude to Dr. Cedric Gondro, whose insightful ideas and initial guidance were instrumental in shaping the foundation of this thesis.

A special thank you goes to my wife, Dr. Sara Bano. Her constant presence, understanding, and encouragement has been a source of strength during the toughest times.

My gratitude extends to my family and friends, who, despite the distance, have always supported me. I hold dear the memories with my friends Dr. Ali Munir and Dr. Muhammad Zeeshan. Their kindness and deep understanding of life, left a lasting impact on me. Many thanks to my colleagues at Michigan State University. In particular, I would like to thank Dr. Yasir Nawaz for his help throughout my PhD journey.

I extend my heartfelt thanks to all the members of our lab, including Dr. Ken Reid and Dr. Iliya Miralavy for their unwavering support, collaborative spirit, and invaluable assistance throughout the course of this research. Your encouragement has made this journey both enjoyable and fruitful.

Finally, I honor the memory of my late parents and extend my deepest gratitude to my sisters. Their love and support have been my foundation, and I dedicate this achievement to them.

TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION	1
1.1	Background	1
1.2	The Case of Beef Supply Chain	8
1.3	Research Statement	13
1.4	Major Challenges and Proposed Solutions	15
1.5	Terms Used in the Thesis	21
1.6	Note on Publications	23
1.7	Conclusion	23
CHAPTER 2	LITERATURE REVIEW	25
2.1	The Supply Chain Network	25
2.2	Collaboration in Supply Chain Networks	27
2.3	Modeling a Supply Chain Network	29
2.4	Regulations in Food Supply Chains	32
2.5	Organizational Functions in the Beef Supply Chain Network	33
2.6	Traceability in Food Supply Chains	34
2.7	Digital Ledgers for Food Supply Chains	36
2.8	Conclusion	39
CHAPTER 3	OVERVIEW OF THE BEEFMESH COLLABORATION FRAMEWORK	41
3.1	System Functionality	41
3.2	System Requirements	44
3.3	Blockchain Consortium Infrastructure	45
3.4	Connecting Distributed System Components	48
3.5	Initiation and Formation of Collaboration Groups	49
3.6	Conclusion	56
CHAPTER 4	IMPLEMENTATION DETAILS OF BEEFMESH FRAMEWORK	57
4.1	Software Tools and Custom Scripts	57
4.2	Implementing the Blockchain Application Layer	58
4.3	Managing Authorizations in a Collaboration Group	60
4.4	Data Pipeline for Consuming Localized Data	61
4.5	Supporting Off-Chain and On-Chain Common Data	63
4.6	Maintaining Traceability with Data Splitting	66
4.7	Knowledge Transfer Pipelines for Traceability and Collaboration	67
4.8	Enabling Trust Through Hardened Framework Security	69
4.9	Conclusion	71
CHAPTER 5	TRACEABILITY EXAMPLE AND SYSTEM EVALUATION	73
5.1	Traceability Example	73
5.2	Data Logging for Traceability	74
5.3	Testing System Performance	79
5.4	Direct Traceability Applications	87

5.5	Conclusion	87
CHAPTER 6	TRACKING CARBON FOOTPRINT USING BEEFMESH FRAME- WORK	89
6.1	Basics of Carbon Emissions	89
6.2	Related Work on Emissions	92
6.3	The Carbon Tracking Application	93
6.4	Internet of Things as the Enabler for Emissions Tracking	95
6.5	Results and Discussion	106
6.6	Conclusion	111
CHAPTER 7	OPTIMIZING RESOURCE UTILIZATION USING BEEFMESH FRAME- WORK	112
7.1	Introduction	112
7.2	The Resource Consumption Optimization Application	114
7.3	Results and Discussion	117
7.4	Conclusion	129
CHAPTER 8	ENABLING SECURE KNOWLEDGE TRANSFER PIPELINES US- ING BEEFMESH FRAMEWORK	130
8.1	Introduction	130
8.2	The Federated Learning Data Pipelines Framework	137
8.3	Securing Federated Learning Data Flow Channels	139
8.4	Example Applications and Discussion	140
8.5	Conclusion	152
CHAPTER 9	CONCLUSION AND FUTURE DIRECTIONS	154
9.1	Summary of the Thesis	155
9.2	Supported Applications and Future Directions	156
9.3	Limitations of the Proposed Framework	157
	BIBLIOGRAPHY	160

CHAPTER 1

INTRODUCTION

The existing issues in modernized supply chains have led to a disconnected and fragmented supply chain network with less vertical integration and no information sharing. This has resulted in scenarios of supply chain security threats, exploitation by major corporations, limited traceability, lack of transparency and reduced trust for collaboration between organizations. This chapter summarizes the prevailing issues in complex supply chains along with the need for a decentralized framework that enables client-controlled collaborative applications. At the end, the foundation and motivation for a decentralized and distributed collaboration framework is laid out to solve the prevailing issues. A summary of the major challenges in building the proposed framework along with the implemented solutions is also presented.

1.1 Background

The recent COVID-19 pandemic revealed several weaknesses in supply chain systems globally with numerous reported cases of broken supply chains and limited availability of most of the daily consumed commodities [1, 2]. Most noticeable technology issues included the lack of digital information related to supply chain activities; the lack of infrastructure to securely share critical information (e.g., delay in delivery) with participants; and the lack of infrastructure to extract and share possible insights from existing common data to predict and prevent future issues [3]. Security threats for supply chains also increased over time due to uncontrollable counterfeit products and increased cyber-attacks [4]. Agricultural production was among the top supply chains affected

by these issues due to the complex connection of subsystems from production to sales, with only limited information shared through fixed point-of-sale channels [5].

The quantity and complexity of data generated from disjoint participants throughout the supply chain restricts sharing of information due to the underlying privacy concerns [6]. The complexity of the data comes from the use of numerous complex subsystems, that includes testing laboratories, integration systems and the use of sensors that collect and store data from the internal supply chain functions. Industrialized supply chains now integrate data sources that generate data in petabytes or higher. For example, affordable Deoxyribonucleic Acid (DNA) sequencing of cattle can result in large amounts of data in the meat industry. Organizations in the supply chain that lack the necessary equipment and infrastructure required to process and store large amounts of data generated daily, eventually end up discarding it [7]. Even with the advancement of 'Big Data' technologies, a large section of organizations are still hesitant to incorporate it mainly due to the lack of expert resources to implement and manage it. With the increasing number of devices connected to the Internet, there is always room for a compromise in security that results in reduced trust in adoption of latest technology. A properly integrated and digitized supply chain system can therefore, provide the first step towards privacy-preserving knowledge sharing. A connected supply chain can help collaboratively managing material scarcity at different states of the supply chain, counter uncontrolled price hikes, predict unavailability of freights, solve traffic congestion issues by forecasting market demands and incorporate consumer response that can pave way for supply chain transparency.

The agricultural supply chain requires most attention because food is consumed daily, based on the trust that it is created following food safety regulations indicated by food labels. Take the case of '*Beef Supply Chain*' for example, which requires more accountability due to numerous stakeholders and its high global impact from carbon emissions and food borne diseases [9, 10, 11]. The beef supply chain network, not only involves numerous stakeholders, but also carries global consequences because of being shipped worldwide in large quantity. Our daily diets increasingly include red meat, poultry, pork, and seafood, which is a particular concern due to their association

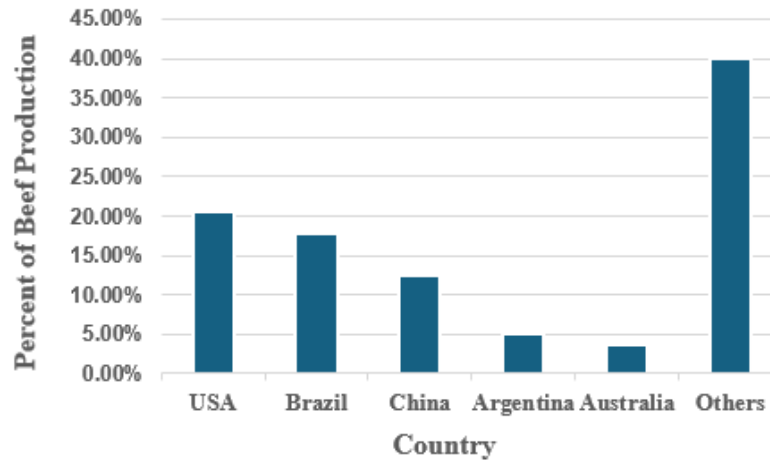


Figure 1.1 United States and Brazil take the largest share of world's beef supply [8]

with diseases such as E. Coli, Bovine Spongiform Encephalopathy (BSE), Salmonella, Scrapie, and Trichinosis, when linked to tainted meat [9]. Red meat alone represents a multi-trillion-dollar industry, surpassing the oil sector in terms of economic scale [10]. Companies involved in the beef supply sector, like National Beef, Tyson Foods, JBS, and Cargill, process vast quantities of beef daily, distributing it locally and globally through intricate supply chain networks (as shown in Figure 1.2). This growth trend is expected to continue, with top producing countries such as USA, Brazil, China, Argentina, and Australia contributing in the expansion (as shown in Figure 1.1).

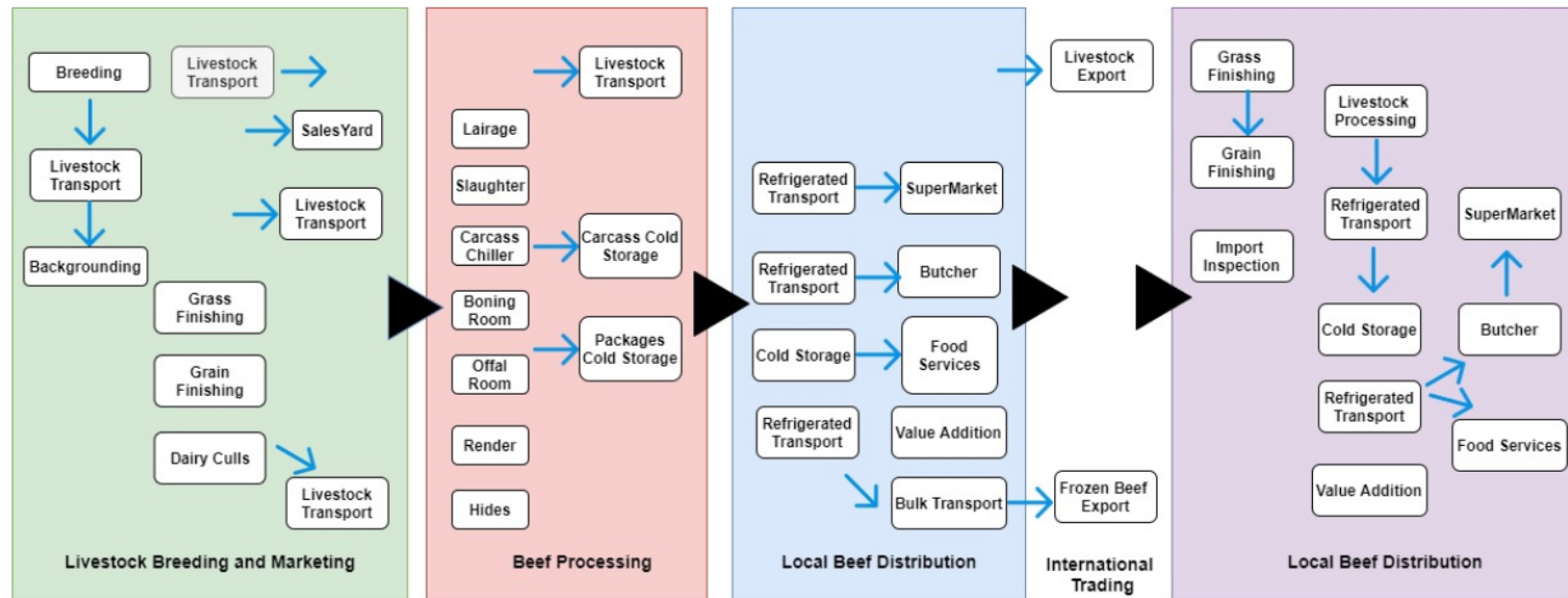


Figure 1.2 Beef supply chain network is a complex system that involves numerous stakeholders with different trade agreements [12]

With a staggering demand of beef globally for 7.8 billion people, organizations in the beef supply chain are looking to transition to a digital ecosystem. But a scalable technology to record, track, collaborate and regulate the industry involving all underlying participants is lacking. The inability to transfer timely vital information (e.g., disease outbreaks) incentivizes fraud and allows larger companies to create monopolies [13]. On a broader level, the issues in current food supply chain systems (with focus on the beef chain) can be summarized as: (i) minimal use of vertical integration (ii) lack of data transparency and traceability (iii) outdated scalable digital ledger for storing diverse information from multiple user domains (iv) lack of infrastructure for inter-organization communication [14, 15, 16].

With independent subsystems along the beef chain, reliably tracking, reporting and managing critical information from *'farm-to-fork'* is a challenging task. This is made further difficult by the fact that staggering supply chain demands have pushed organizations to produce more output at the expense of adopting complex subsystems. The demand for animal protein has increased cattle raising and mechanisms to move cattle quickly through the supply chain has resulted in creating enormous negative environmental impact [11]. The modern versions of the beef chain now incorporate complex industrialized subsystems that include managing livestock, animal feed harvesting, meat processing plants, cold storage, transportation and retail stores. This complexity comes with major issues such as the lack of mechanism to reliably record, extract and track vital information before data is lost or discarded. Even if useful information is extracted locally from the tremendous amount of data generated in each subsystem, most of it is lost or overwritten due to the lack of means and mechanisms for collaborating and sharing data with other organizations. Even if there were a common database for sharing data, mutually controlling and regulating it is a complex task since it requires defining numerous policies beforehand. Knowledge transfer in a complex supply chain with disjoint participants is therefore, particularly challenging because data is collected and stored with local and private jurisdictions. Sharing of data, that mutually benefits all participants, therefore requires continuous infrastructure expansion with different data communication channels. Building such a framework would require addressing organizational

concerns summarised in Figure 1.3.

A number of solutions in literature have been proposed to regulate and share data in supply chains but each has a number of shortcomings that limits its adoption. Blockchain based frameworks serve as popular ledger systems to store and trace data but incorporate a number of limitations that are summarized in Table 1.1 [17, 18, 19, 20, 21, 22]. Blockchain limitations mainly originate due to the way it is designed and integrated with other applications, opening the door to cyber-attacks. These attacks have been reported to manifest themselves as functional flaws when the blockchain is coupled with other resources on the same network including off-chain databases and applications[23, 24, 25]. Since the blockchain on its own is fairly limited as a digital ledger for storage, it requires integration with other applications to allow data regulation, thus opening possible vulnerabilities. A compromise in the integration process can put data consistency, control, privacy, confidentiality and user acceptability in jeopardy [24]. In the literature, adoption of blockchain approaches, particularly for consumable supply chains, claims to improve transparency [26], allow reliable data collection [20, 27, 27], prevent information tampering [28], reduce data tracking complexities [29], improve transportation [30] and provide numerous incentives to participants [31]. The blockchain in general and particularly in agricultural applications, however, has been reported to suffer from the same limitations discussed in Table 1.1.

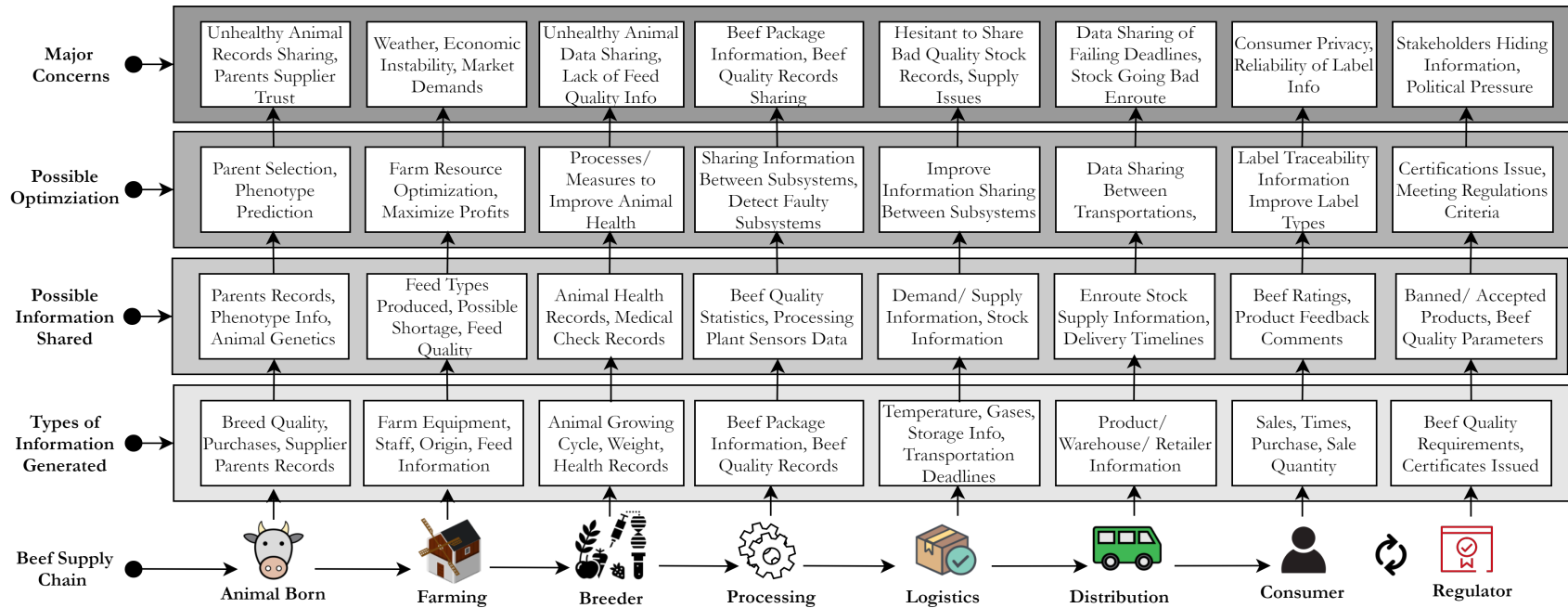


Figure 1.3 Different subsystems in a ‘Beef Supply Chain’ system generate diverse data that can be leveraged by a connected information platform for sharing useful knowledge and mutually optimizing processes by addressing underlying concerns

1.2 The Case of Beef Supply Chain

The ‘Beef Supply Chain’ represents a classic example of an extremely complex chain which limits knowledge transfer due to disjoint operation, has transparency concerns and numerous data federation limitations arising from its current industrialized layout. Though it is difficult to summarize all of the processes in a beef chain from end-to-end [32], a simplified mathematical model can be defined in the following way,

Given:

t Time index (e.g., days, weeks, months)

I_t Inventory of beef at time t

P_t Production of beef at time t

D_t Demand of beef at time t

C_t Capacity constraints in the supply chain at time t

C_{prod} Production cost per unit

C_{inv} Inventory holding cost per unit

The inventory dynamics and production constraints are:

$$I_{t+1} = I_t + P_t - D_t$$

$$P_t \leq C_t$$

An objective function can be defined as:

$$\text{Minimize } \sum_t (C_{\text{prod}} \cdot P_t + C_{\text{inv}} \cdot I_t)$$

The variable parameter ‘demand’ D_t in the above system of equations can refer to several scenarios. If the demand is modeled considering retailer as the end of the supply chain, then it would directly translate to the amount of beef that needs to be shipped. If the final stage of the supply chain is considered as the consumer, then demand will directly translate to the amount of beef that is sold

Current Limitations	Consequence for Supply Chains
Use of central control authority	Lack of trust; single failure point; limited data control
Storing large datasets directly in blockchain	Slower transactions with increasing consensus nodes; not feasible for massive data sets; frequent consensus time outs
Use of permissionless blockchain	Exposed transactions; exposure to malicious nodes
Use of non-configurable network infrastructure	Inability to form multiple participant tiers (sub-groups) for collaboration and information sharing
Poorly configured decentralized applications prone to disruptions	Affects reliability of information (e.g., traceability); affects scalability for multi-tier connections
Limited bidirectional end-to-end information flow	Inability to resourcefully optimize chain locally and globally; inability to predict instability in operations
Limited reconfiguration of framework	Limited applications for sharable data and limited timely information dissemination
Fixed information flow interfaces	Inability to flexibility accommodate chain participants and changing data use-cases
Costly blockchain transactions	Less motivation for supply chain participants to use blockchain for traceability of internal chain functions
Proprietary and costly solutions	Limited and expensive applications for supply chain; less flexibility for custom multi-tier applications

Table 1.1 Limitations of current work in connecting disjoint supply chain participants using blockchain based platforms

and consumed. Considerations such as transportation, storage, processing times, perishability, and market conditions may require additional variables and constraints in a more comprehensive model. Specialized inventory control algorithms may be applicable to the beef chain once information from major parts of the complete supply chain network is readily available for processing at some common node. Inventory control algorithms are critical for efficiently managing stock levels and optimizing supply chain performance [33]. These algorithms include the Economic Order Quantity (EOQ) model, Just-In-Time (JIT) inventory, and the Newsvendor model, each addressing different inventory management aspects. The EOQ model focuses on minimizing total inventory costs by finding the

optimal order quantity that balances ordering and holding costs [34, 35]. JIT inventory aims to reduce holding costs by closely aligning orders with production schedules, thus minimizing excess inventory [36, 37]. The Newsvendor model deals with uncertain demand, balancing overstocking and understocking costs [38]. Advanced techniques like multi-echelon inventory optimization further enhance supply chain efficiency by optimizing safety stock levels across multiple stages of the supply chain [39].

A major challenge in gathering of information from majority of the organizations in the beef supply chain, for inventory modeling or for other reasons, comes from its disjoint and complex nature. Another challenge is the lack of a consolidated infrastructure connecting all of the beef supply chain participants that can facilitate collecting shared data and in transfer of knowledge. At a broader level, major issues in the beef chain can be summarized by its disjoint nature, transparency and its distributed layout inhibiting connectivity through currently used technology. To demonstrate the usefulness of our proposed collaboration framework in this thesis, we implement an example supply chain application (described in detail in Chapter 5) that focuses on the beef supply chain network. Some important aspects of the beef supply chain that shapes the motivational foundation for the application developed in the thesis, are therefore summarized below.

1.2.1 Disjoint operations in the beef supply chain network

Modern versions of the beef supply chain now incorporate complex industrialized subsystems that includes, but are not limited to, livestock management, animal feed harvesting, meat processing plants, cold storage, transportation and retail stores. For decades, these subsystems have worked independently, yet input of one heavily depended on the output of others. This allowed larger enterprises to dominate the most profitable portions of the chain [40, 41]. This form of disjoint operation in the beef supply chain is a major reason for counterfeit products, fraud and inability to improve productivity, sustainability and traceability [16, 42, 43].

1.2.2 Trust and transparency issues behind the beef supply chain

Among all meat chains, the beef supply chain industry in particular requires more traceability, accountability and information sharing, not only due to the numerous stakeholders involved, but

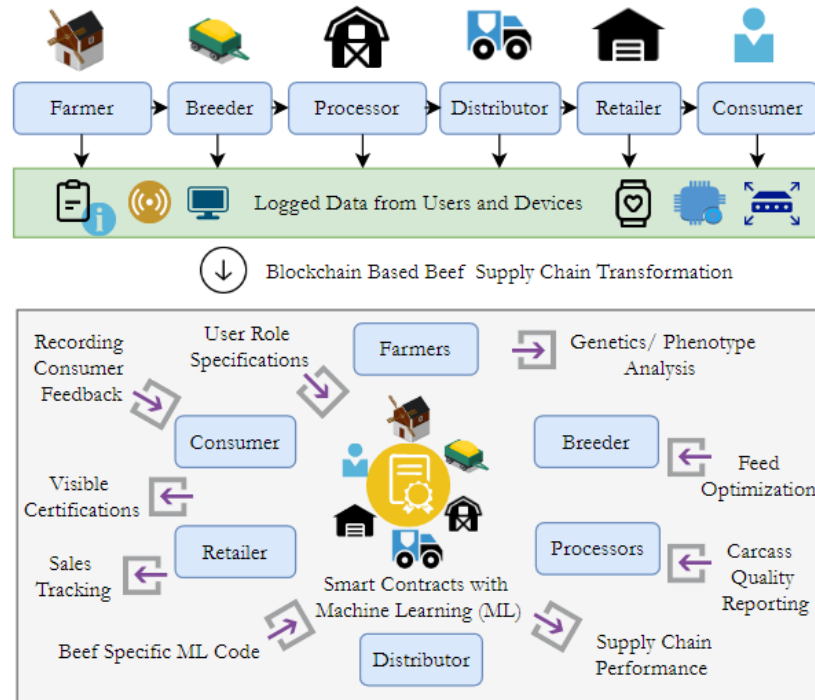


Figure 1.4 Use of blockchain and distributed applications in the beef supply chain network can facilitate numerous benefits for participants

also due to its global impact. Its global impact includes massive forest destruction, degradation of land, disappearance of water reserves, over fertilization, unbalanced biodiversity and poor air quality [44]. A reliable traceability system that takes into account all stakeholders would result in acceptance by consumers for beef quality, contamination tracking, monitoring contagious diseases and advocacy for humane animal treatment [45, 46, 47, 48, 49, 50, 51].

1.2.3 Current technology limiting knowledge transfer

The beef industry generates massive amounts of data ranging from animal genetics information to supply chain processing. Only a part of this information is stored or shared. Major sources of vital data include animal testing laboratories, genetics information, animal harvesting factories, storage and transportation systems [52]. A major portion of data is discarded due to limited resources. For example, a bovine genome can take up more than 3GB of storage space consisting of 3 billion nucleotide pairs, and the data contains approximately 22 thousand genes [53]. Despite major advancements in big data technologies, studies show that beef supply chain decision makers

are hesitant to directly use shared common data, for fear of privacy concerns and trust issues [54]. Conventional techniques for capturing data such as Near Field Communication (NFC), Radio Frequency Identification Devices (RFIDs), Wireless Sensor Networks (WSNs), Bluetooth Low Energy (BLE) and Global Positioning System (GPS) still require filtering, screening and processing data for security and privacy reasons before transferring it to other jurisdictions. The use of a centralized ledger system adds further difficulty in terms of data control and single point of failures. Use of blockchain alone in the beef chain does not satisfy the heavy requirements of the beef chain industry, thus requiring the use of off-chain databases, authentication servers, data routing devices and Internet of Things (IoTs)/sensors [23, 24].

1.2.4 Summary of technology issues faced by the beef supply chain

Provisioning an end-to-end beef supply chain communication and collaboration functionality is still in its inception, not only because of its disjoint and distributed operation but also because of the lack of technology that can address the following issues:

1. Fragmented working and individual control of parts of the beef supply chain which results in trust issues.
2. Lack of integration of data processing and knowledge consolidation tools and techniques to take in data and convert it to useful information from processes and events occurring at massive scale.
3. Lack of infrastructure and data pipeline to securely generate, store and share traceability information that is thoroughly regulated by privacy policies.
4. Lack of infrastructure to scale and integrate digitally connected supply chain participants without disruption.
5. Lack of infrastructure and integrated techniques to control connected supply supply chain participants from threats, for example cyber attacks and flow of counterfeit items.

6. Lack of a infrastructure pipeline to provide timely and secure access to common knowledge generating data and its transfer.
7. Lack of means and mechanism to jointly use and enable machine learning applications over common data to globally optimize supply chain.

1.3 Research Statement

Motivated by the lost opportunity of collaborative knowledge transfer from disconnected supply chain participants, we propose and implement a framework around blockchain and other distributed services that mitigates the limitations highlighted in Table 1.1.

In this thesis, we look at the end-to-end supply chain connectivity issues from a holistic view and provide a generic consolidated solution to the problems originating from supply chain fragmentation. The research statement therefore addresses the question:

“How to design and implement a generic, scalable and reliably connected end-to-end digital supply chain collaboration network for sharing common knowledge?”

There are four main critical components of the research statement, namely: (1) A generic solution (2) A scalable design (3) A reliable network and (4) A data-driven framework.

A ‘*generic*’ framework means that the system allows arbitrary number of participants with distinct roles to join or leave seamlessly and connected collaboration groups for distinct applications to be formed or destroyed without disruptions. A ‘*scalable*’ framework means that the system uses a modular approach for arbitrary number of formations (e.g., a group or consortium of organizations) to be created and connected in a decentralized manner without the continuous need for a centralized authority. A ‘*data-driven*’ framework means that the system is able to reliably store and retrieve particular data for participants when required and disseminate it to authorized entities through end-to-end bidirectional information flow channels. Finally, a ‘*reliable*’ connectivity framework is guaranteed by a disruption free, secure and privacy preserving end-to-end bidirectional communication setup. The proposed framework can be reconfigured for different applications on

the fly with consensus from the participating organizations. These four requirements pave the way for a secure and privacy preserving information sharing platform that enables trust, traceability and transparency in disjointed supply chains. The above mentioned requirements ultimately translate to development of reliable beef supply chain collaboration applications and operations.

Keeping in view the research statement defined above, we propose, implement and demonstrate a secure, scalable and traceable supply chain connectivity and collaboration framework built using an integration of blockchain and distributed application services that can be easily re-configured for any underlying collaboration task. The framework, among other applications and features, can be used for tamper-proof traceable common data keeping at any level, while being able to scale to various degrees in a decentralized manner. With a federated, distributed and decentralized architecture approach for data keeping and knowledge transfer, the system enhances organizational trust, policy and decision policy making capabilities while automating regular supply chain tasks. In short, we propose and implement a re-configurable application framework developed by seamlessly integrating distributed resources (nodes, databases, services and interfaces) that enables configuring different types of secure data communication channels between closed local or global collaboration groups. Such a decentralized and distributed, yet highly connected framework of supply chain collaboration groups allows building numerous applications such as reliable tracking of tradeable commodity, optimizing supply chain resource consumption and jointly applying Machine Learning (ML) on federated data for knowledge extraction. Earlier works in this direction either collected data manually from different participants of the chain with numerous assumptions, or extracted it from only a limited section of the supply chain to present a collective analysis of shared knowledge. In addition, methods proposed earlier relied heavily on central application servers which are considered a potential source of discouragement for collaboration between disjoint participants considering the sensitivity of data and private organizational jurisdictions. At the end, a collaboration application for knowledge sharing is more appreciated and favorable for supply chains when the participants have full control over their own data, the underlying technology, the form of collaboration groups and the channels on which information flows.

1.4 Major Challenges and Proposed Solutions

A decentralized supply chain connectivity framework for distributed participants to form collaboration groups and jointly improve traceability, negotiate policies, implement decisions and regulate data for sharing knowledge comes with numerous challenges. Main challenges and their proposed and implemented solutions that are described in the thesis are summarized below.

1.4.1 Fixed data flow channels

With a fixed ‘point-of-sale’ channel between pairs of organizations to communicate on, it is difficult to negotiate policies, decisions and to start joint projects around shared knowledge. To address this issue, we configure and provision the collaboration framework with private, public, and hybrid blockchain channels, integrating both local and global distributed databases and services, to cater to a wide range of applications. This comprehensive approach ensures seamless, secure, and efficient data management and service delivery tailored to diverse needs of federated data sharing.

1.4.2 Dispersed common knowledge data sources

A major challenge in building our collaboration framework is to non-intrusively select and interface numerous potential knowledge generation data sources operating in jurisdiction of private organizations. These data sources (data feeds) are scattered throughout the supply chain and generate potential useful information for the same product (processed beef in our example application). To address this issue, we leverage a hierarchy of networked sources, each with varying levels of access and connectivity, to ensure robust data management. The framework integrates databases governed by private jurisdictions to enhance the security and efficiency of common data applications, providing tailored solutions for specific legal and regulatory environments.

1.4.3 Limited blockchain storage capability

The blockchain in itself is fairly limited for use in the supply chain network because frequently storing and retrieving large datasets is expensive and allowing participants to jointly manage it results in a complex network layout. To overcome these challenges, we provision the ability to integrate independently owned (local) distributed resources (*databases, interfaces, nodes* and *ser-*

Major Challenges	Proposed and Implemented Solutions	See Section
Fixed (non-configurable) point-of-sale connections limiting collaboration for other use cases	Provision private, public and hybrid blockchain channels coupled with distributed (local & global) databases and services for different applications	1.4.1, 4.2
Scattered common data (knowledge) sources with private jurisdictions	Utilize tiered (hierarchical) and privately connected (networked) databases for common data applications	1.4.2, 4.2, 4.5
Limited storage capability of blockchains	Securely coupling blockchain with off-chain shared and distributed databases	1.4.3, 4.5
Information splitting (forking) in the supply chain as cattle moves from end-to-end	Embed common information within other related information to form metadata that points to original data	1.4.4, 4.6
A large amount of data generated in varying formats from internal supply chain events, functions and processes	Map data to GS1 formatted codes after data intake (processing) from data format compatible interfaces before saving in relevant databases	1.4.5, 2.6, 5.2.2
Countless traceable data parameters in the beef chain	Utilize a predefined beef chain domain specific database with distinct parameters	1.4.6, 5.2
Time sensitive data that cannot be stored on blockchain	Integrating time stamping authority to establish validity of data against time sequence	1.4.7, 4.3
Tiered characteristics of organizations and users in supply chain with different privacy restrictions	Utilize a re-configurable blockchain framework with communication (collaboration) channel groups at consortium, organization and sub-organization level	1.4.8, 3.5
Organizations with limited hardware, software and other capabilities	Provision containerized resources including databases, services and applications	1.4.9, 3.5.1
Keeping network and collaboration group up and running to avoid application downtime and disconnections	A mutually managed redundant server (collaboration point) maintains essential resources (information, files, configurations) to keep groups running and collaborating	1.4.10, 3.5.2
Legitimate and authorized users in a group sabotaging data and manipulating information flow	Enforce privately networked distributed databases using data application nodes spread over a secure consortium to enable data redundancy and replication	1.4.11, 4.8

Table 1.2 Summary of challenges and proposed solutions for the supply chain collaboration framework

vices) that can be started and seamlessly integrated into the collaboration network as containerized applications. This allows to utilize off-chain data storage, redundancy for critical data, access to data from a variety of sources using appropriate interfaces, preserving tiered (hierarchical) data control (federation) and employing secure data communication channels for collaboration.

1.4.4 Splitting of data at different stages

Considering the case of the beef supply chain, large amount of data with common (relevant) knowledge for participants is generated at each step but due to the disjoint nature of organizations, it is difficult to track continuous information for animals. Data splitting (*forking*), e.g., an animal in a meat supply chain processed into thousands of consumable packages, poses data mapping and data consolidation challenges. Another difficulty in this scenario is to establish the timeline and scope of trace information in a timely manner. To solve issues prevalent to the data context, data forking (splitting), scope of trace timeline, and data explosion, we make use of off-chain shared databases distributed along the supply chain. This allows embedding context within other data to form metadata. The shared distributed database is local and global at the same time in the sense that it helps maintain traceability for the whole chain by maintaining portions of data at different nodes. By negotiating and defining data privacy policy in a group through communication channels, globally maintained data is authorized to be viewed by selected users.

1.4.5 Diverse formats of captured data

A functionality is needed that can capture relevant data and its associated metadata in different scopes (e.g., time, location) with potential of generating knowledge. We make use of industry standard Global System of Standards (GS1) codes [55] to capture and convert triggered events into associated traceable unit metadata. With different types of databases configured to run locally and globally and part of a federated permissioned consortium of the beef chain network, any form of data such as information from laboratories, reproduction facilities, feeding houses, logistics, meat quality certification agencies, carcass examiners and weather reporting stations can be captured and shared where required [56].

1.4.6 Wide array of recordable parameters in the beef supply chain network

Traceability of events, processes, functions and transactions in a supply chain work like a railroad system where each block represents a change happening at some point, and contents inside the block hold details of the changes done. It becomes a challenge to decide relevant parameters of interest and the method to record them particularly if events are triggered during a short period of time. We make a distinction in our framework between *event*, *process* and a *function* in the sense that an event is the result of a process and a process could involve several functions. For example in a meat (beef) supply chain, processing machines perform various tasks (functions) to gradually cut down meat (a process) that results in a final consumable tagged meat package (an event). Taking the case of beef supply chain, we define a handful of predefined domain parameters that are collected at various stages of the supply chain, e.g., cattle health and feed related parameters are defined for the breeder organization (refer to the specific use of the term ‘breeder’ in Section 1.5). A number of configured databases and interfaces allow consuming information in the form of streaming data, high volume sensor data, static data and in other formats (e.g., data feed).

1.4.7 Validating non-blockchain time-sensitive data

It is not feasible to store large files directly on the blockchain due to the time required for consensus and the potential for data explosion. Therefore, an integrated off-chain database is necessary for file storage. This integration necessitates a method to verify the authenticity of files and their timestamps, as the sequence of file changes cannot be reliably established using the blockchain alone. To solve this issue, a timestamp authority service is utilized during logging large data files (e.g., cattle genetics data) for cross verification and for verifying time related tasks that cannot be validated from blockchain.

1.4.8 Hierarchical relationship of organizations

Another challenge for developing a collaboration framework for supply chains lies in provisioning the application to be re-configurable so that it can be customized for each organization (locally) or group without disrupting any global goals. Considering the beef supply chain as a tightly controlled multi-organization setup, deciding on the number of permissioned (or public)

distributed database nodes (e.g., blockchain nodes), their role and ownership for sharing knowledge from data is a difficult task. To solve this issue, we decided to use a hybrid permissioned (blockchain) consortium as one of our frameworks layer by making a decision from a number of options for our application requirements (as shown in Figure 3.2). A major challenge here, in bringing together diverse organizations for collaboration, is to define and implement policies for access rights to information generated from resources (e.g., IoTs and sensors) in different jurisdictions. Data cannot flow outside of a closed organizational setup until a decision is made on the nature of information to share, the organizations to share them with and the technological mechanism to enable data sharing. The beef supply chain with disconnected participant organizations, including farmers, breeders and processors represent a classic example of this scenario. To solve these underlying issues, our proposed framework allows forming a re-configurable collaboration group at consortium, organization and sub-organization levels. Communication channels within each group are used for different collaboration applications, e.g., to provide traceability or for sharing policy related to a project. Each group, after connecting, can decide to divide the group into sub-groups to allow limiting access of data sets to certain sub-groups. Any sub-group with access to a ledger can further enforce policies that restrict data viewing rights to a user group. Hence, our proposed application allows creating a multi-tiered (hierarchical), disruption free infrastructure that works around federated group resources to allow filtering information as it is passed along. Several key software level protection policies further enforce protection of sensitive data in the proposed application.

1.4.9 Organizations with limited hardware and software capabilities

A limitation in bringing participants together for collaboration and using the same software application layers is the varying underlying hardware capabilities. Some organizations may lack efficient hardware resources with ample storage to run resource-intensive applications effectively. To solve this issue, we provision lightweight containerized resources, including databases, services, and applications, to build the collaboration framework. These containerized applications are flexible enough to be configured according to the specific needs and the underlying hardware resources.

1.4.10 Ensuring continuous operation of the collaboration framework

The collaboration network initiates and expands through collaboration forking points, allowing the network to grow and scale. Therefore, a minimum set of services must be available to serve users at certain collaboration points, enabling organizations to form groups. A mutually managed redundant server (collaboration point) maintains essential resources such as information, files, and configurations to support continuous group collaboration. At a bare minimum, a single blockchain node with one application channel and a single distributed database node is sufficient to initiate and grow multiple collaboration groups. This node can be maintained by a Non-Government Organization (NGO), a regulatory 3rd party, or as a mutually managed node run by participating organizations.

1.4.11 Disruption from authorized users

When connecting disjoint supply chain organizations to allow sharing common data between them, security concerns may arise for scenarios where authorized users may sabotage a running application by injecting malicious data. When data moves across different organisations with aggregation of information at each stage, manipulation of data becomes the most common issue since actual stakeholders of data get removed from the control plane. For example, a malicious animal breeder group during the time of cattle sale, can decide to share manipulated genetics data of cattle with processor organization to gain momentary sale or purchase incentives. To allow consistent control of data by the organization that originally generates it, our application uses distributed database nodes with each stakeholder organization controlling at least one node. The databases are connected through private networking to jointly maintain common data and any information appended to it. Hence with data redundancy and replication benefits from peer-to-peer functionality, data is prevented from being manipulated in any collaboration group. The private network that connects and manages databases is configured using each nodes Internet Protocol (IP) address that is securely shared during group formation. Hence our application allows maintaining data integrity for localized setups (e.g., breeders mating their animals regionally by sharing animals statistics) or for global applications that require maintaining end-to-end product related trace data

across the supply chain.

1.5 Terms Used in the Thesis

The challenges and implemented solutions for our proposed framework are particularly applicable for complex supply chains such as the ‘beef supply chain’. We therefore refer to the ‘beef supply chain’ network throughout the thesis when providing domain specific example scenarios. The example application implementation that we demonstrate in the thesis, built around the beef supply chain scenario for collaborative tasks, is referred to as the “**BeefMesh**”. A number of other terms have been used in the thesis, particularly within the specific context of a beef supply chain network. These terms are defined below for reference.

Breeder: A *breeder* in the thesis refers to an individual, a group of individuals or an organization responsible for managing the mating of animals to produce offspring. To make the beef supply chain scenario simpler, we include the role of a feeder within the breeder organization. Hence, the breeder organization’s tasks also include, arranging feed for the animals and ensuring their proper care and management during the breeding process. Additionally, in the context of this thesis, breeders can potentially raise animals from birth to full growth. In reality, the relationship between a breeder and a feeder within the beef supply chain is sequential and complementary. The feeder is an entity that sits in between a breeder and abattoir and is responsible for managing the nutrition and growth of the cattle, typically providing a high-energy diet to fatten them up to the desired market weight before they are sent for processing. A breeder in the context of this thesis, however, includes the operations of a ‘feeder’ and a ‘farmer’ within it.

Client: The term *client* in a scenario utilizing our proposed framework, corresponds to the ‘user’ of an application or service that the framework provides.

Collaboration Group: A *collaboration group* is a set of participants that mutually enable different applications by utilizing shared common data and network infrastructure. These groups work together to achieve collective goals by leveraging shared resources and information.

Consortium: A *consortium* is a set of *organizations* or a set of participants with a common goal, e.g., a farmer, transporter, and retailer collectively recording traceability data.

Distributor: A *distributor* is the entity responsible for shipping meat packages or live animals between different parts of the supply chain. This role includes utilizing cold storage and other logistic functions to ensure the safe and efficient transportation of products from processors to retailers or from other sources to particular destinations.

Event: An *event* is defined as the outcome of a process. For example, in a beef supply chain, an event could be the final consumable tagged meat package.

Farmer: In the context of a beef supply chain network in the thesis, a *farmer* refers to an organization responsible for bringing in calf, raising them, and growing feed for the cattle to consume. The farmer's role includes all activities related to the care and feeding of cattle until they are ready for further processing. Since this is synonymous with the definition of the term 'breeder' that we defined earlier, we make a distinction between the two in terms of animal mating. Particularly, we use the term 'breeder' more frequently and define it to include the operations of a 'farmer' as well as a 'feeder'.

Feeder: In the context of a beef supply chain network, the word *feeder* used in the thesis refers to an individual or organization responsible for providing feed and managing the nutrition of cattle. Feeders ensure that animals receive appropriate diet to support their growth and health, often during specific stages of their development before they are sent for further processing to abattoir.

Function: A *function* refers to the specific tasks or operations performed by an entity (e.g., an electronic device) in the supply chain that contributes to the overall process of moving a product from end-to-end. For example, in a beef supply chain, the processor 'function' refers to the overall process of cutting down beef into smaller pieces.

Group: A *group* is a collection of participants within an *organization* sharing common goals among themselves and with the *organization*, which can be different from another *group* in the same *organization*.

Organization: We define *organization* as a set of supply chain participants with unique goals that does not overlap with other *organizations*, e.g., an independent farmer or a set of breeders.

Participant: The term *participant* is interchangeably used to represent an individual user (or client)

of a supply chain application. The term participant can also refer to any of the described formations (organization and group) utilizing the connectivity framework for a specific application.

Process: A *process* can involve several functions. For instance, in a beef supply chain, processing machines perform various tasks (or functions) to gradually cut down meat, which is considered a process.

Processor: In the context of the supply chain network, a *processor* refers to an abattoir organisation, responsible for slaughtering cattle and processing the beef into consumable products. This role includes the tasks of butchering, packaging, and preparing meat for distribution to retailers or consumers.

Retailer: In the context of the beef supply chain, a *retailer* is the entity responsible for processing meat into packages of different sizes and then placing these packages in retail stores for consumers to purchase.

Supply Chain Network: We define *supply chain network* as the interconnected network of organizations, individuals, activities, information, and resources involved in the process of moving a product or service from suppliers to end customers.

1.6 Note on Publications

Content from chapters of this thesis, in part or in full, is under preparation for submission (or has been submitted) for publication. The inclusion of this content in the thesis is in accordance with the university policies on thesis publication. This note is included to ensure there is no conflict when the content is subsequently published in conferences and journals before or after the thesis is published.

1.7 Conclusion

The supply chain network currently suffers from fragmentation, which is compounded by increasing complexity due to modernization. This fragmentation hinders effective communication among participants beyond fixed regional points-of-sale. Although organizations within the supply chain share common data, they face significant restrictions. A framework that addresses the underlying issues of privacy, data control, and flexible collaboration channels could therefore

unlock numerous potential applications. In this chapter, we summarized the main challenges arising from fragmentation in the supply chain, emphasizing the need and motivation for a collaborative framework. This chapter lays the foundation and proposes solutions to overcome these challenges, particularly in the context of the beef supply chain network. By addressing the specific issues and presenting targeted solutions, we aim to create a cohesive, consolidated and efficient supply chain collaboration framework.

CHAPTER 2

LITERATURE REVIEW

The thesis proposes a decentralized and distributed supply chain collaboration framework which extensively incorporates concepts from food supply chain, food regulation, traceability, beef supply chain and blockchain. A brief summary of these topics and related work is presented in this chapter.

2.1 The Supply Chain Network

The concept of the supply chain has evolved significantly over the past century, driven by technological advancements, globalization, and changing market demands. In the early 1900s, supply chain systems operated largely as independent, fragmented components. Each function, such as manufacturing, warehousing, and transportation, worked in isolation, leading to inefficiencies and high operational costs [57, 58]. This period was characterized by limited communication and coordination between different components of the supply chain.

The period between the 1970s and early 1980s saw a gradual shift towards consolidation of various supply chain functions. Key processes like packaging and warehousing began to merge, creating more streamlined operations [58, 59] (as shown in Figure 2.1). This era marked the beginning of recognizing the benefits of a cohesive supply chain network, although full integration was still a distant goal.

The start of the 21st century brought about a revolution in the supply chain industry, primarily due to the rapid advancement in Information Technology (IT). The introduction of sophisticated IT solutions facilitated real-time data sharing and communication across different components

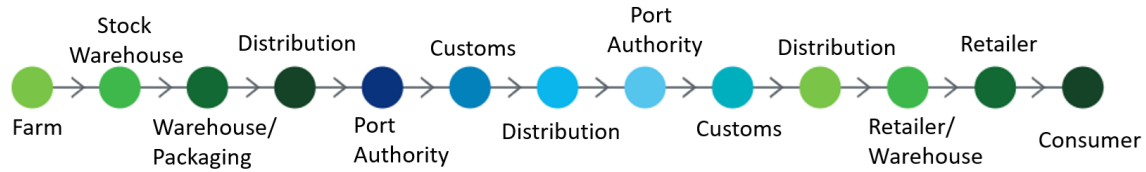


Figure 2.1 The supply chain network involves various participants including farms, warehouses, manufacturing plants, distribution centers, and retail outlets, each playing a crucial role in the flow of goods and information

of the supply chain, enabling more efficient and responsive operations [60]. Technologies such as Enterprise Resource Planning (ERP) systems, Electronic Data Interchange (EDI), and Radio Frequency Identification (RFID) played crucial roles in transforming supply chain management into a more integrated system. These technologies enabled companies to track inventory, forecast demand accurately, and optimize logistics, significantly reducing costs and improving customer satisfaction [61].

The vision of Industry 4.0 has further transformed supply chain networks by leveraging autonomous, seamlessly connected machines operating within smart factories. These machines communicate and coordinate with each other, optimizing manufacturing processes and reporting all activities to a central commanding authority. This interconnected setup enhances efficiency and adaptability within the supply chain, enabling businesses to respond swiftly to market demands [62]. Industry 4.0 incorporates advanced technologies such as the IoTs, Artificial Intelligence (AI), and big data analytics, which facilitate predictive maintenance, automated decision-making, and real-time optimization of supply chain processes within a participating organization [63].

The evolution of supply chain networks also reflects a growing emphasis on sustainability and resilience. Modern supply chains are pushing towards the minimization of environmental impact through the adoption of green logistics practices, such as optimizing routes to reduce fuel consumption and implementing eco-friendly packaging solutions [64]. With advancement in technology, there is a push for supply chains to be engineered in a way that allows it to be more resilient, capable of withstanding disruptions caused by events such as natural disasters, geopolitical tensions, and pandemics [65].

In conclusion, the evolution of the supply chain network from fragmented components to inte-

grated systems and now towards autonomous smart factories and blockchain-enabled transparency reflects significant advancements in technology and management practices. These changes have enabled supply chains to become more efficient, responsive, resilient and adaptable to the dynamic demands of the global market. Although technology has revolutionized supply chains, making them more efficient and interconnected, however, information sharing remains a complex and challenging task due to factors such as lack of vertical integration, monopolistic practices and concerns of data privacy.

2.2 Collaboration in Supply Chain Networks

The foundation of supply chain collaboration lies in the willingness of various entities within the supply chain to share information and work towards common goals. Effective collaboration necessitates a robust technological platform that enables communication and connectivity among different players and functions within the supply chain. Such technological connectivity is essential for real-time data sharing, coordination, and decision-making, which are crucial for efficient and responsive supply chain operations [66].

The timeline of significant milestones in the development of supply chain collaboration highlights the progressive integration of technology and practices that have transformed supply chain management. In 1913, Henry Ford's introduction of the automobile production assembly line marked a significant advancement in manufacturing efficiency, paving the way for future collaborative practices in the supply chain [67]. The 1950s saw the introduction of shipping containers, which revolutionized logistics and transportation by standardizing cargo handling and reducing shipping times and costs [68].

The issuance of the first patent for barcoding in 1952 was another crucial development, enabling automated tracking and management of inventory [69]. This innovation was further enhanced by IBM's introduction of the IMPACT platform in 1967, which aimed to digitize inventory management and streamline supply chain operations [70]. The Universal Product Code (UPC), introduced in 1974, standardized product identification, facilitating more efficient and accurate inventory tracking and sales processing [67].

The late 1970s saw the popularization of Electronic Data Interchange (EDI) devices, allowing for the electronic exchange of business documents between organizations. This significantly improved the speed and accuracy of information sharing [71]. In 1983, the ARPANET project connected hundreds of computers, laying the groundwork for the modern internet and enhancing collaborative capabilities in supply chains [72]. Walmart's introduction of the cross-docking system in 1985 exemplified the use of real-time data to optimize inventory management and reduce storage costs [73].

The 1990s introduced the concept of lean production, promoted by Toyota, which emphasized efficiency, waste reduction, and continuous improvement in manufacturing processes [74]. In 1995, the Collaborative Planning, Forecasting, and Replenishment (CPFR) methodology was developed, promoting joint planning and information sharing among supply chain partners to improve forecasting accuracy and inventory management [75]. The use of Electronic Product Codes (EPC) in RFID technology in 1999 further enhanced the ability to track products throughout the supply chain, providing greater visibility and traceability [76].

The 2000s witnessed significant advancements in non-line-of-sight scanning tools and the widespread use of the internet to communicate with customers. These technological advancements enabled real-time tracking of shipments and enhanced customer service [77]. These innovations have been pivotal in transforming supply chain collaboration from isolated activities into a seamless, integrated process.

In conclusion, the evolution of supply chain collaboration has been marked by significant technological advancements and the gradual integration of various functions within the supply chain. From the early days of the assembly line to the modern era of digital connectivity and real-time data sharing, these developments have enabled supply chains to become more efficient and responsive to meet the dynamic demands of the global market. In short, although collaboration in the supply chain has taken various forms and its practice has become increasingly visible, it remains a complicated process due to the complexity of underlying organizational structures and concerns over data privacy.

2.3 Modeling a Supply Chain Network

Modeling a supply chain network involves creating a structured representation of the entire supply chain, including all entities and their interconnections. Based on the requirements, supply chain models can be categorized into the following types [66]: (1) Continuous flow models that maintain stable processes over time (2) Fast chain models designed for products with short life cycles, emphasizing speed and responsiveness (3) Efficient chains that focus on optimizing efficiency to remain competitive (4) Custom-configured models tailored for products, featuring re-configurable processes to accommodate unique requirements (5) Agile models suitable for specialty products that require flexibility and quick adaptation to market changes and (6) Flexible models that operate with continuously evolving processes to adapt to dynamic environments.

When modeling a supply chain network, it is essential to consider both horizontal and vertical dimensions, which include the tiers, actors, and 3rd party collaborators. The decision parameters for such models are defined by constraining them to a manageable number of outcomes, which then determine the performance metrics and objective functions to be used. For example, decision parameters might include allocated resources, network structure, actors or stages in the supply chain, sequence of services, workforce size, and the level of outsourcing or 3rd party involvement.

2.3.1 Categorization of supply chain models

Supply chain models are essential tools for analyzing and optimizing various aspects of supply chain operations. Addressing different aspects of supply chain management, these models can be broadly categorized into several types: (1) Deterministic models (2) Stochastic models (3) Dynamic models (4) Network models (5) Hybrid models, and (6) IT-driven models.

Deterministic models assume that all parameters and variables are known with certainty. These models are particularly useful in stable environments where variability is minimal, allowing for precise planning and optimization. Linear Programming and Integer Programming are common examples of deterministic models, providing solutions for resource allocation and production scheduling [66, 78]. Most frequently used techniques in deterministic models category include the use of Linear Mixed models, Fuzzy Logic, heuristics, meta-heuristics, Genetic Programming

and Stochastic techniques. Other applied techniques include Fuzzy logic, Linear, Non-Linear models and continuous approximation. Example use cases of deterministic approaches in supply chain include the modelling of supply chain participants, environmental considerations, social interactions, business competition, constrained decision making, planning strategies and scheduling resources [78].

Stochastic models, on the other hand, account for uncertainty and variability in supply chain parameters. These models are crucial for decision-making in environments where demand, supply, and lead times are unpredictable. By incorporating probabilistic elements, stochastic models help managers develop robust strategies to mitigate risks and manage variability. Examples of stochastic models include Probabilistic inventory models, Markov processes with probabilistic differential equations and simulation models [66, 79]. Major sub-categories of stochastic techniques used in supply chain include Dynamic Programming and Control Theory.

Dynamic models consider the time-dependent behavior of supply chains, capturing the evolution of the system over time. These models are essential for understanding changes in inventory levels, production rates, and other time-sensitive factors. Dynamic Programming and System Dynamics models are typical examples, enabling the analysis of temporal changes and their impact on supply chain performance [66, 80].

Network models represent the supply chain as a network of nodes (e.g., suppliers, manufacturers, warehouses) and arcs (e.g., transportation links). These models are used to optimize the flow of goods and information through the supply chain network. Transportation models and network flow models are common examples that help in designing efficient logistics and distribution systems [66, 81].

Hybrid models combine elements of deterministic, stochastic, dynamic, and network models to address complex supply chain problems. These models provide comprehensive solutions by capturing multiple dimensions of supply chain operations. Hybrid models are particularly useful in scenarios that require a balanced approach to manage different aspects of the supply chain simultaneously [66]. Some popular techniques in this category include the use of Mixed Statistical

models, Information Rule based models, Petri-nets, Echelon Structural modeling, Mixed Integer Programming, Integrated Structural Modeling and Confirmatory Factor Analysis in addition to combined use of Integer Programming with fuzzy approaches [66].

Lastly, IT-driven models leverage advanced information technologies to enhance supply chain performance. These models utilize data analytics, machine learning, and artificial intelligence to optimize supply chain processes. Predictive Analytics models and Decision Support Systems are examples of IT-driven models that enable real-time decision-making and improve overall efficiency [66, 82]. Since our proposed supply chain collaboration framework falls under the IT-based solutions category, we therefore, summarize in section 2.3.2, existing solutions and, in particular, frameworks closely resembling to our application.

In summary, each supply chain models offers unique advantages for addressing specific challenges in supply chain management. By understanding and applying these models, businesses can optimize their operations, mitigate risks, and enhance overall performance.

2.3.2 Information technology driven models and frameworks

Over the years, several simulation and analysis tools have been developed, categorized under IT-driven models. These tools facilitate the understanding of complex processes and interactions involved in large supply chains, along with the management of various network functions. For instance, *Arviem* (<https://arviem.com/>) serves as a transportation management tool. *DiCentral* (<https://www.dicentral.com/>) provides order processing solutions for supply chains and offers lean inventory management systems. *Fishbowl Inventory* (<https://www.fishbowlinventory.com/>) is a tool for warehouse management. *Freightos* (<https://www.freightos.com/>) offers a framework for freight handling services. *Coupa* (<https://www.coupa.com/>) is used for bidding in supply chains, while *Softeon* (<https://www.softeon.com/>) is a popular tool for managing supplier functions. The SAS (https://www.sas.com/en_us/industry/retail/solution/demand-planning.html) platform is widely used for demand forecasting in supply chains. *Palo Alto Networks* (<https://www.paloaltonetworks.com/>) provides a security framework for supply chains, and compliance auditing tools include platforms like *MetricStream* (<https://www.metricstream.com/>).

Numerous IT-based startups have emerged within the agricultural supply chain industry over the years. For example, *TraceFood* (<https://tracefood.io>) is a framework that provisions a traceability application by storing data in blockchain. Although application like *TraceFood* and similar blockchain based initiatives claim to be decentralized because of the underlying blockchain technology, however in reality, they cannot be considered fully decentralized because of a number of reasons. This is so because the blockchain resources and all other integrated software applications required to provision traceability or other functions reside on servers controlled by the company. Hence, in practice, majority of the control over the application and software resources resides with 3rd party, that results in trust issues over data control. A partially decentralized framework, where the majority of control resides with the solution provider, assumes that all participants agree with the central authority to provide static data for provenance. However, this framework does not offer a generic end-to-end multi-point bidirectional connectivity interface mechanism that can scale to different proportions, integrate data sources, and provide multiple application potentials. Moreover, storing large data chunks on the blockchain is not feasible due to the requirements for processing and validating transactions by other participating nodes. This is particularly true for majority of the currently available traceability solutions in the market, examples of which include the *TE-FOOD* (<https://te-food.com>), *CABCattle* (<https://cabcattle.com>), *AgriDigital* (<https://www.agridigital.io>), *EthicHub* (<https://www.ethichub.com/es>), *Ripe.io* (<https://www.ripe.io>), *OriginTrail* (<https://origintrail.io>), *BlockYard* (<https://www.blockyard.com/>), *TraceX* (<https://tracextech.com/>) and *Performance Live-stock Analytics* (<https://www.performancelivestockanalytics.com/>). A summary of the applicable limitations of these solutions is also summarized in Table 1.1.

2.4 Regulations in Food Supply Chains

The increase of distance that consumable food travels from production stage to the final consumer product has given rise to the importance of maintaining food safety [83, 84, 85]. Food can get contaminated as a result of the introduction of some physical agents, by some chemical reaction or by biological means [86]. Hence a number of food safety regulations are in place by standardizing agencies around the globe. For example in the USA, the Food Drug Administration (FDA) and the

United State Department of Agriculture (USDA) regulate food safety standards [87]. The European agency mandating food safety measures is called the European Food Safety Authority (EFSA) [88]. These are just examples from a few countries. The food regulatory authority in Australia maintains its own safety system called National Livestock Identification System (NLIS) [89]. A generic principle for all food safety regulations in supply chains mandates maintaining a minimum level of traceability of records [90].

The European Union (EU) was the first to take an initiative along with 45 countries to mandate traceability across supply chains by introducing the Global Standard (GS) [91]. This provided guiding principles for food industry to attach traceability information in the form of encoded digital strings on their products at various stages by using available (wired or wireless) technology [92]. Along with the GS1 standard, the FDA also released the Food Safety Monitoring Act (FSMA) in 2011 to prevent further health risks in food processing [93]. After EU and FDA regulations, a number of frameworks came out for use in supply chains. One of the earliest of these frameworks includes the International Organization for Standardization (ISO) 22005:2007 traceability system [94, 95]. Some other standards meant to standardize supply chain processes include the ISO 12875, ISO 12877 and ISO 22095 [47, 96].

2.5 Organizational Functions in the Beef Supply Chain Network

As we advocate our supply chain collaboration framework with the example of beef supply chain, a summary of the underlying stages is presented. The beef supply chain system begins with a cattle farmer starting and maintaining a calf lot. Calf are mostly fed with pasture grass in their early days up until 6 months of age, after which they are weaned at the feeder lots. At the feeder (refer to the use of terms ‘feeder’ and ‘breeder’ in Section 1.5) lots, they are fed a variety of grain mixed diets. Once this breeding phase called ‘finishing’ is over, the cattle are either sold at auction or sent to beef processing facilities (abattoirs). From the processing facilities, beef packages end up in retail stores across the country are exported globally through distribution companies to be ultimately bought by consumers. As cattle move from one end to the other, hundreds of critical and potential common knowledge parameters are generated at various steps of the beef chain. A

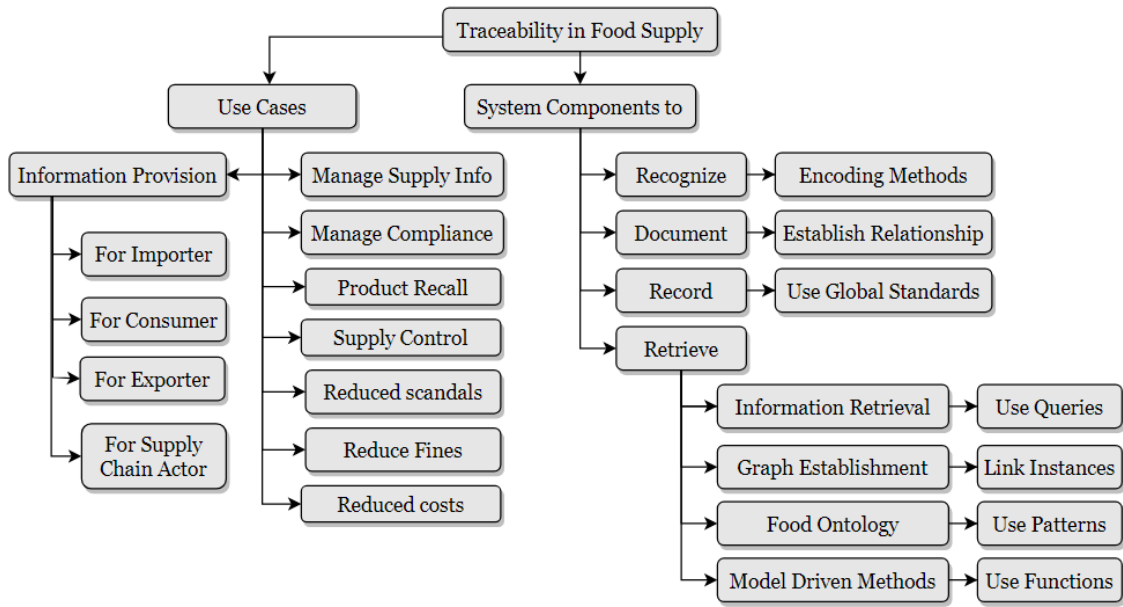


Figure 2.2 Traceability in food supply system can be broadly defined in terms of its use case and the systems (applications) that enable it

handful of such important parameters used in the our proposed framework are described in Chapter 3 and listed in Table 5.1. In general, a beef supply chain network can be broken down into breeder (with feeder included) functions involving feed production and cattle raising, processor functions involving animal harvesting and beef packaging, and transportation functions involving cold storage [97, 98, 99]. More details of the beef supply chain network are highlighted in Section 1.2 or can be accessed at the USDA platform (<https://www.usda.gov/meat>). A reference to the use of terms in this thesis, particularly in context of the beef supply chain network, is provided in Section 1.5.

2.6 Traceability in Food Supply Chains

Traceability in agricultural production is the ability to fetch information related to a part of the supply chain network, (e.g., retailer, processing or supply) when required. ISO 22005:2007 defines traceability as the capability to trace the path of a product through the supply chain [100]. At a broader level, traceability in food chain systems can be defined in terms of the underlying application use and the technological means and mechanisms to achieve it (as shown in Figure 2.2). An ideal implementation of a traceable system solves the problem of: (1) the data type that needs to collected (2) the ownership of data at each stage (3) the type of equipment required to collect

and store data (4) the interpretability of data and (5) the ability to include or exclude any external or internal participants in the supply chain when required [101, 102].

With the advancement in technology for data generation and collection, agricultural companies have started to deploy applications like RFIDs, Wireless Sensor Network (WSN), Bar codes, Quick Response (QR) codes and Internet of Things (IoT). Radio Frequency Identification Device (RFID) based techniques allow each food item in the supply chain to be tracked and recorded at various stages of the supply chain using electromagnetic waves or cheap RFID tags [103, 104, 105, 106, 107, 108]. A major issue with RFID based methods are the difficulty to correctly read information due to disruptions from signal collisions [109, 110]. QR codes and bar codes have been the main methods to digitize, record and track agricultural items for decades in supply chains [111, 112, 113]. Near Field Communications (NFCs) also work on the same principles as RFIDs but use a different communication technology [114]. QR codes and bar codes are printed as machine readable tags to store information for food items. Major issues with the digital codes to track items include the reliability of data. The information attached with any form of code can be changed after it is generated [115]. Digital codes on their own cannot be used to collect information and require other means to collect data that is then transformed into compact codes. IoT and WSNs represent two different classes of monitoring systems where intelligent sensor systems are deployed across the supply chain to automatically gather and report information [101, 116, 117, 118, 119, 120]. Major concerns with IoT based systems in agricultural supply chains are related to undesirable access, lack of sophisticated encryption and privacy leaks [121]. Major issue with deploying WSNs in agricultural supply chains are the cost of devices, high bandwidth requirements for communication and disruptions from external attacks [122].

Traceability in the food supply chain is essential for maintaining food safety, enhancing consumer trust, and efficiently tracking food recalls. The GS1 standard, initiated by EU, is a comprehensive framework that helps in identifying, capturing, and sharing information about products, assets, services, and locations in a standardized manner [92]. The GS1 standard helps with supply chain management and traceability by providing guidelines to convert data into standard codes

summarized below.

1. The Global Trade Item Number (GTIN) is a unique identifier for trade items, enabling consistent and accurate tracking of products at various packaging levels throughout the supply chain.
2. The Serial Shipping Container Code (SSCC) identifies logistic units, such as pallets or containers, facilitating the tracking and tracing of these units to ensure accurate and efficient logistics management.
3. The Global Location Number (GLN) identifies physical locations, such as farms, processing facilities, and distribution centers, and is crucial for tracking the production, processing, and storage of products.
4. The Global Data Synchronization Network (GDSN) allows trading partners to share and synchronize product data, ensuring access to accurate and up-to-date information, which is essential for effective traceability and supply chain management.
5. The Electronic Product Code Information Services (EPCIS) standard enables the sharing of information about the movement and status of products in the supply chain, providing details on where a product has been, what has happened to it, and its current location.

Our proposed application framework utilizes the GS1 standard codes to enhance traceability in the beef supply chain. Utilizing these standards within our supply chain collaboration application offers a comprehensive solution for traceability and operational efficiency in the beef supply chain network.

2.7 Digital Ledgers for Food Supply Chains

Technology such as IoT, WSNs, RFIDs, NFCs and barcodes only represent means and mechanisms to generate, collect and report traceability information. A system is still needed to allow integration of subsystems that would typically be running their own separate Supply Chain Management (SCM) tools. Several digital ledger systems for data management have been proposed

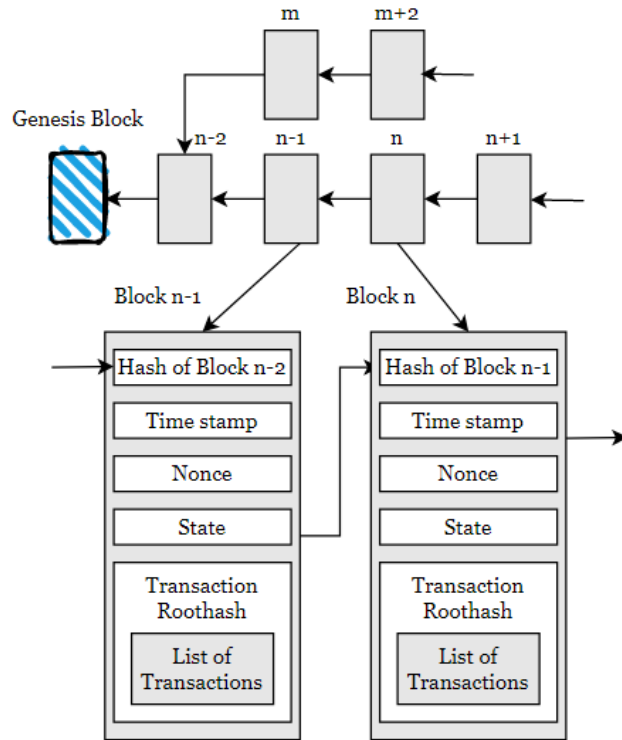


Figure 2.3 Basic building block of a blockchain network

but all come with their shortcomings in terms of security, privacy, data control and scalability [123]. The current major types of technologies for Digital Ledger Technologies (DLTs) include blockchain, Directed Acyclic Graph (DAG) and Hashgraphs. Blockchains work under the principle of consensus, meaning all transactions in the network need to be validated by participants of the network [124]. Approved changes in the network are then recorded as chains of information blocks (as shown in Figure 2.3) with a time sequence that can help establish traceability. A major issue with blockchain is the fees, in the form of tokens, required to perform transactions. Other issues with blockchain include its complexity of use and the difficulty to integrate it with other technologies. DAG based DLTs work by the principle of sharing information by forming a network that takes the form of a DAG [125]. Such a system can scale to large sizes but is limited by centralized nodes. Hashgraphs work by forming a combination of consensus and gossip network [126]. Hashgraphs only work with permissioned networks and due to the patented nature of the underlying methods and therefore, has not been thoroughly investigated for agricultural supply chains.

Blockchain as a decentralized ledger, utilizes a number of functions to allow storing and retriev-

ing records [127]. A fundamental building block in a blockchain is a cryptographic hash function. Let H be a hash function applied to a data block X , the hash function can then be represented as $H(X) = Y$. Given two hashes A and B , the blockchain makes use of merkle trees for efficient verification of data integrity against A and B , which is calculated as $MerkleRoot(A, B) = hash(A + B)$. To agree on storing a record in a distributed manner, blockchains make use of a consensus algorithm. Although, consensus algorithms can vary, the most commonly used algorithm ‘proof-of-work’ involves finding a Nonce (N) such that the hash of the block $headerH(blockheader + N)$ meets certain criteria. Finally consensus is determined by using data blocks and network state as $Consensus = f(current\ block\ data, previous\ block\ data, network\ state)$. In addition, blockchains also make use of asymmetric key pairs (public key, private key).

Adoption of blockchain for the agricultural supply chain in the literature claims to improve transparency [26], prevent information tampering [28], reduce data tracking complexities [29], improve transportation [30] and provide incentives to participants [31]. Cao et al. proposed a framework to allow collecting data using sensory devices installed at various stages of the beef supply chain network [27]. The work however, is based on a preliminary study and lacks a concrete implementation. Similarly, some other work in the space of beef chains have provided preliminary studies on the use of smart contracts [128] and consumer demands [129], but lack a concrete implementation. An application for the ‘beef chain’ related data management has been described in the work by Tanvir et al., but lacks details on its scalability and implementation [20]. The shortcomings of this work include missing beef chain specific smart contracts, use of off-channel communication, centralized servers and use of Relational Database Systems (RDSs) alone, which does not support functions like sharding.

Several attempts have been made to leverage the benefits of IoT, RFIDs and NFCs in blockchain networks for traceability. Aung et al. combined RFIDs, NFCs and sensory networks for tracking quality and shipment status of food items [130, 131]. Rajeb et al. combined IoT devices with blockchain to enhance the value, efficacy and effectiveness of the supply chain processes [132]. Mondal et al. proposed to integrate custom RFIDs, IoT and blockchains by dividing it into physical

and cyber layers [133]. Off-chain data storage for blockchain is required because of the computational complexity, scalability and the size of data generated from supply chains [134]. Baralla et al. proposed a system to monitor critical information from cold chains in agricultural supply systems by integrating IoT and off-chain data storage [135]. Huang et al. utilized the InterPlanetary File System (IPFS), which is a distributed peer-to-peer storage mechanism, to store data generated from IoT embedded with Electronic Product Codes (EPCs) [136]. Leng et al. experimented with a multi-chain architecture developed over blockchain to solve the problem of communication between disjoint systems in a supply chain [137]. Khaled et al. developed a blockchain based soybean traceability system that incorporated IPFS to record immutable information for reliable provenance [138].

Recently, major technology related companies have also started to show interest in the development of blockchain based solutions. Companies like *Walmart* and *IBM* are testing pilot programs to ensure meat safety by enabling traceability [139]. Other noticeable brands working on traceability in food supply chains include *Merck*, *Baidu*, *Auchan*, *Accenture*, *KPMG* and *Carrefour*, *Maersk*, *British Airways*, *UPS*, *Nestle*, *Unilever*, *JD.com*, *InterAgri* and *FedEx*. Startups trying to solve interoperability issues in blockchain based supply chains to improve provenance have been discussed in Section 2.3.2, where the majority of these platforms are limited some of the shortcomings summarized in Table 1.1.

2.8 Conclusion

In this chapter, we explored the multifaceted nature of supply chain networks, emphasizing the critical role of collaboration in enhancing efficiency and integration across various segments. The application of theoretical frameworks and modeling approaches from literature was discussed with a view of their effectiveness in optimizing operations. Emphasis was placed on the importance of regulatory compliance to maintain food safety standards, while specific organizational functions within the beef supply chain were examined to address sector-specific challenges. A collaborative framework controlled by participants to enable applications like traceability and provenance was identified as a pivotal element, particularly in food supply chains where trust, transparency and

consumer safety are paramount. The transformative potential of digital ledger technologies, such as blockchain, was discussed for their ability to enhance traceability and data integrity. This chapter illustrated that a combination of strategic collaboration, adherence to regulations, adoption of advanced technologies, and effective modeling is essential for building resilient, transparent, and efficient supply chains capable of adapting to different demands.

CHAPTER 3

OVERVIEW OF THE BEEFMESH COLLABORATION FRAMEWORK

This chapter provides a description of the collaboration framework, covering its overall structure, functionalities, and the distinctions between its various components. Since the example application is designed and implemented around the particular scenario of a beef supply chain network, we refer to it as the “BeefMesh”. This chapter explores how the different components of the BeefMesh framework function individually and in conjunction with one another to create a seamless and effective system for extracting and sharing knowledge from common data. With a focus on the interactions between different modules and their contributions to the proposed framework’s goals, a comprehensive description of the system’s architecture, the dynamics of its operations, and the integration of its diverse elements is presented.

3.1 System Functionality

A secure, scalable and traceable blockchain based beef supply chain system collaboration framework, named BeefMesh, is proposed that allows building applications around common participants data. Direct applications of the framework include data logging, information extraction, tracking of cattle data from end-to-end, supporting secure federated learning pipelines and knowledge transfer between disjoint organizations. Communication of policies and decisions for collaboration is done by forming a consortium group with common blockchain application channels, distributed resources (nodes, databases, services) and interfaces (ports, addresses) for connectivity. An example consor-

tium group can include multiple organizations such as farmers, breeders, processors, distributors, retailers, regulators and consumers where each organization decides on their own to join or leave a group at any time without disruption. An optional administrator (regulator or maintainer) can also be configured easily within a group to allow minimum maintenance of active blockchain channels and distributed databases from which the group can start and scale up any time. Nevertheless, once a group with a particular application is formed and active, it can independently continue working without the requirements of any central authority. Some of the important components of the framework (blockchain and distributed databases) in an example group in our framework are shown in Figure 3.1.

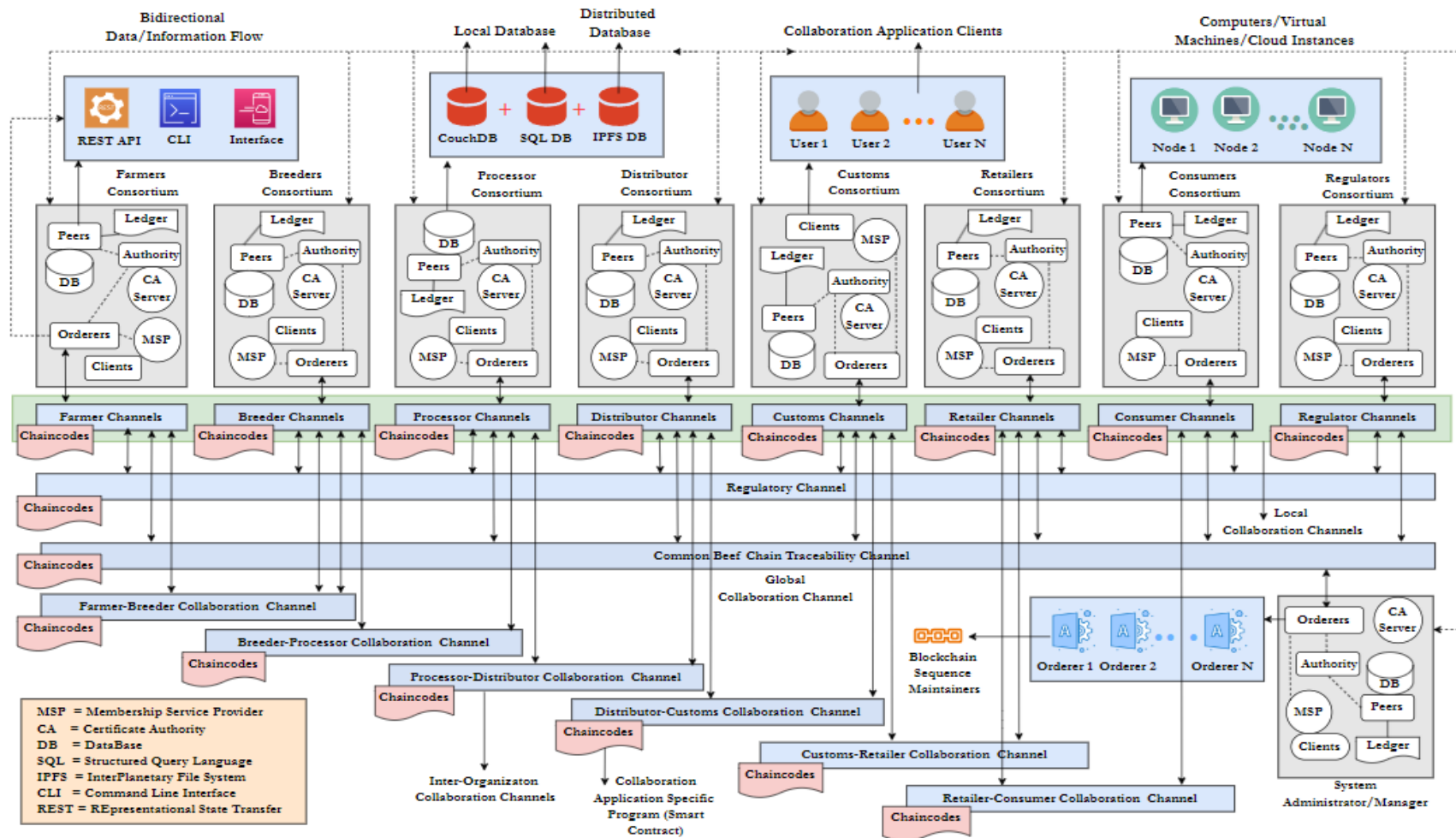


Figure 3.1 A cross section of the example beef chain collaboration group utilizing blockchain and distributed components. Distributed resources utilizing federated nodes and services running on privately configured addresses are networked together using a combination of overlay and bridge networks to allow seamless flow of data, decision making and knowledge transfer

The proposed collaboration framework can be divided into different functional components. At the *Data Level*, system can be broadly categorized into: (1) Control Layer (2) Transaction Layer and (3) Storage Layer, while at an *Application Level* the proposed system can be categorized into: (1) Application Programming Interface (API) Layer (2) Extension Layer and the (3) Protocol Layer. More details of these layers and their functions are described in Section 3.5.3. Overall, the proposed framework simulates the entire process of beef production from ‘farm-to-fork’ while simplifying the process of vital information sharing between main participants. Traceability in the system is provided by recording unique identification number of cattle and mapping it to the beef packages after harvesting. The traceability information is stored using IPFS [140] which is a peer-to-peer based distributed file hosting platform that ensures longer life, privacy and immutability of recorded information. The Content Identifier (CID) (cryptographic hash of the data) for traceability data from IPFS is stored on blockchain providing a full proof record of sequence of operations. The blockchain service part of the framework utilizes open source Hyperledger Fabric (version 2.5) which is configured through custom scripts to run containerized programs for different beef chain applications [141]. Within each sub-organization of the supply chain network, users are configured to use different roles through distribution of membership authorization certificates that provides different access levels (read, write, modify) to information on blockchain and other databases. Specific beef chain smart contracts (also called chaincodes) are provided as a resource for each collaboration group to install on connected blockchain channels for processing, recording and retrieving data related to collaboration specific tasks and functions. A number of other services (local and distributed databases, IoT and sensor interfaces, time stamping authority) along with application specific federated nodes (e.g., carbon emissions management server) are configured to run as part of the collaboration group.

3.2 System Requirements

We implement the supply chain connectivity and collaboration framework with four major requirements which mandate that the framework should be: (1) Generic (2) Scalable (3) Data-driven and (4) Reliable.

A ‘*generic*’ framework means that the system allows arbitrary number of participants with distinct roles to join or leave seamlessly and connected collaboration groups for distinct applications to be formed or destroyed without disruptions. A ‘*scalable*’ framework means that the system uses a modular approach for arbitrary number of formations (e.g., a group or consortium of organizations) to be created and connected in a decentralized manner without the continuous need for a centralized authority. A ‘*data-driven*’ framework means that the system is able to reliably store and retrieve particular data for participants when required and disseminate it to authorized entities through end-to-end bidirectional information flow channels. Finally, a ‘*reliable*’ connectivity framework is guaranteed by a disruption free, secure and privacy preserving end-to-end bidirectional communication setup. The proposed framework can be reconfigured for different applications spontaneously with consensus from the participating organizations. These four requirements pave the way for a secure and privacy preserving information sharing platform that enables trust, traceability and transparency in disjointed supply chains. The above mentioned requirements ultimately translate to configuration and initialization of reliable beef supply chain collaboration applications and operations.

3.3 Blockchain Consortium Infrastructure

An important part of our framework is the underlying blockchain layout and connectivity around which other distributed resources connect and integrate. The blockchain layout for any collaboration group involves different participants (e.g., breeders and processors) at an organization level forming a connected group (a consortium) as show in Figure 3.1. Within each organization (or sub-organization), data from processes and events are recorded on digital ledgers (local databases) after consumption over data compatible interfaces. Pointers (reference) to vital information (e.g., content identifier) are stored on blockchain for sharing internally or externally. At sub-organization level, we can further create user group partitions where specific information is shared inside the partition but not at an inter-partition level. Information can be controlled and shared at any level after creating and joining groups and enabling communication mechanisms (that include channels, networks, connected interfaces and shared databases). A starting node (collaboration

server) is tasked to coordinate the effort of forming groups initially. The secure Representational State Transfer (RESTful) server allows pooling and distributing required resource information for organizations to start forming groups. All organizations that are part of a group willingly join (after scrutiny) and manage information related to type of network (private, overlay, swarm, bridge, public etc), publicly reachable addresses for shared services (such as IPFS database nodes), blockchain channels to join and information for network shared drive. Group related information can be scrutinized by a regulatory authority or the creator (original owner) of the group before allowing others to join. Members of a group can download custom scripts and files allowing them to run containerized services at their end that connect seamlessly with other group members for sharing common useful data. Once a group is up and running with required number of members, resources and connectivity channels, the collaboration server is not required any more. Hence a group can continuously keep collaborating on different applications in a distributed and decentralized manner.

Once collaborating groups are up and running by connected distributed resources, a number of potential applications can be initiated by the group members. For example, farmers can decide to share cattle related genetic information with breeder organizations (refer to the use of terms ‘farmer’ and ‘breeder’ in Section 1.5). Similarly processors and distributors can decide to share specific information among themselves. At a global level, all participating organizations can decide to share common global beef-related information for traceability and viewing by consumers and regulators. Privacy restrictions of data sharing at different stages of the beef supply chain system naturally result in the requirement for a consortium type blockchain [142] where organizations can control permissions to record, edit or view information at user level. Based on the nature and relationship of organizations in beef supply chain, our proposed application uses a hybrid permissioned consortium type of network (sometimes also called a ‘federated blockchain network’). The deciding factors for using a ‘federated blockchain’ for connecting participants has been summarized in Figure 3.2 while the details of individual components (as shown in Figure 3.1) and the overall implementation of framework is discussed in Chapter 4.

3.4 Connecting Distributed System Components

The proposed framework simulates the entire process of supply chain production from ‘*farm-to-fork*’ while simplifying the process of vital information sharing (e.g., traceability data) and collaboration (e.g., learning from data) at different levels of connectivity. In this work, we make a distinction between *organization*, *consortium* and a *group* as follows. We define *organization* as a set of supply chain participants with unique goals that does not overlap with other *organizations*, e.g., an independent farmer or a set of breeders. A *consortium* is a set of *organizations* or a set of participants with a common goal, e.g., farmer, transporter and retailer collectively recording traceability data. A *group* is a collection of participants within an *organization* sharing common goals among themselves and with the *organization* which can be different from another *group* in the same *organization*. The term *participant* is interchangeably used to represent an individual supply chain application user (client) or any of the described formations (organization and group) utilizing the connectivity framework for a specific application. A summary of the terms used in the

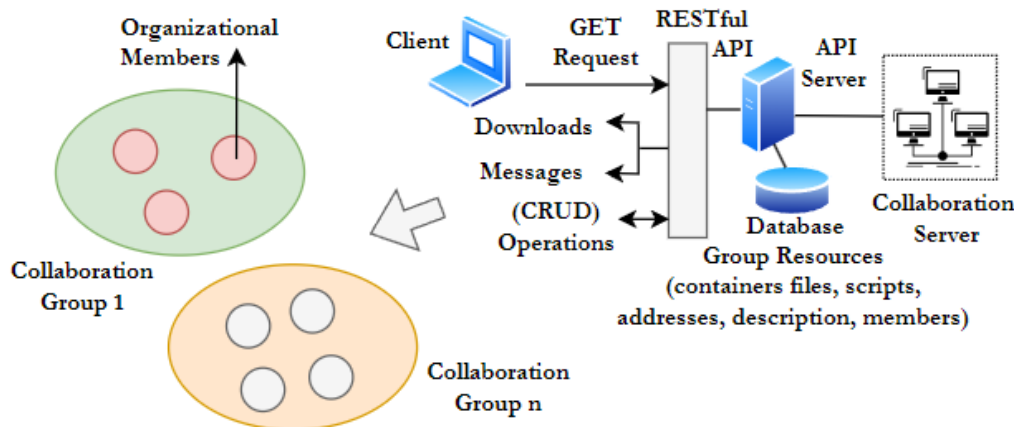


Figure 3.3 Clients (of organizations) interact with a starting collaborator server (also called group initiator) using API calls to register and receive information needed to start forming groups comprised of distributed resources that include database and services with common network connections and data channels

thesis under specific context is also provided in Section 1.5 for reference.

3.5 Initiation and Formation of Collaboration Groups

A RESTful API front-end serves as the starting point for organizations to register themselves in a collaboration group, manage group information (resources), and use the services (containers) provided to build and securely connect with other members in a distributed way. The collaboration initiator application (as shown in Figure 3.3) also serves to authenticate users and groups along with secure management and sharing of group resources. Authenticated users and vetted groups share group related resources in the form of files (with allowed extensions) and data (names, addresses, ports). We experimented with different configurations as the starting point for the collaboration group and ultimately found that the best approach, which avoids privacy issues, group hijacking, and security concerns, is the one which is not fully controlled by any of the participants and is rather mutually managed along with other participating organizations.

To securely connect distributed services (e.g., blockchain, IPFS, shared drive) in a group with overlay networks, a docker swarm manager/worker is used [143]. For local applications (e.g., local database and IoT services), the overlay network is replaced with a local bridge (private) network so that containers can communicate internally. Overlay networks are commonly used in applications like peer-to-peer systems, content distribution, and Virtual Private Networks (VPNs).

Since services run as containerized applications on our platform, an overlay network helps facilitate the communication with other containers without the necessity of configuring complex routing on the individual hosts running docker daemons. A key requirement for achieving this functionality is to ensure that the hosts running the services are part of the same docker swarm. For seamlessly running connected services, files need to be shared among group members (e.g., blockchain channel genesis file). This is achieved by setting up a secure shared drive among members. For starting shared network drive, information related to GlusterFS [144] application is communicated with other members. The shared information includes addresses of hosts and volumes (directory) information. GlusterFS requires at least two servers to host data for redundancy while allowing arbitrary number of clients to join and replicate shared drive. Information necessary to form private IPFS networks among registered and authenticated users is also internally shared through the group initiator server. A private IPFS network is formed by removing all global reachable addresses and allowing only private node addresses to form an IPFS cluster. Encrypted or un-encrypted files can be uploaded to IPFS and both the IPFS content identifier $CID = HashFunction(Content)$ and public key from encrypted data $EncryptedData = Encrypt(OriginalData, PublicKey)$ are shared over separate blockchain channels. The final CID string is obtained by combining the hash and the 'codec' of the data. The codec of data contains the information of the hash algorithm (multihash), mechanism to interpret the hashed data after retrieval (multicode) and a string representation for CID (multibase).

3.5.1 Running connected group services on distributed host nodes

The proposed framework builds upon a distributed permissioned blockchain consortium (as shown in Figure 3.1) which includes stateful (LevelDB) database (also called ledger) for storing blockchain events, a credentials distributing node (Membership Service Provider (MSP)), a Certificate Authority (CA) to validate security related documents and blockchain related transactions processing and sequence ordering nodes (Orderer). The blockchain foundation is further extended by integrating and initializing, connected local databases (PostgreSQL, MySQL, CouchDB, MariaDB and MongoDB) for storing relational and non-relational data while providing a data pipeline for

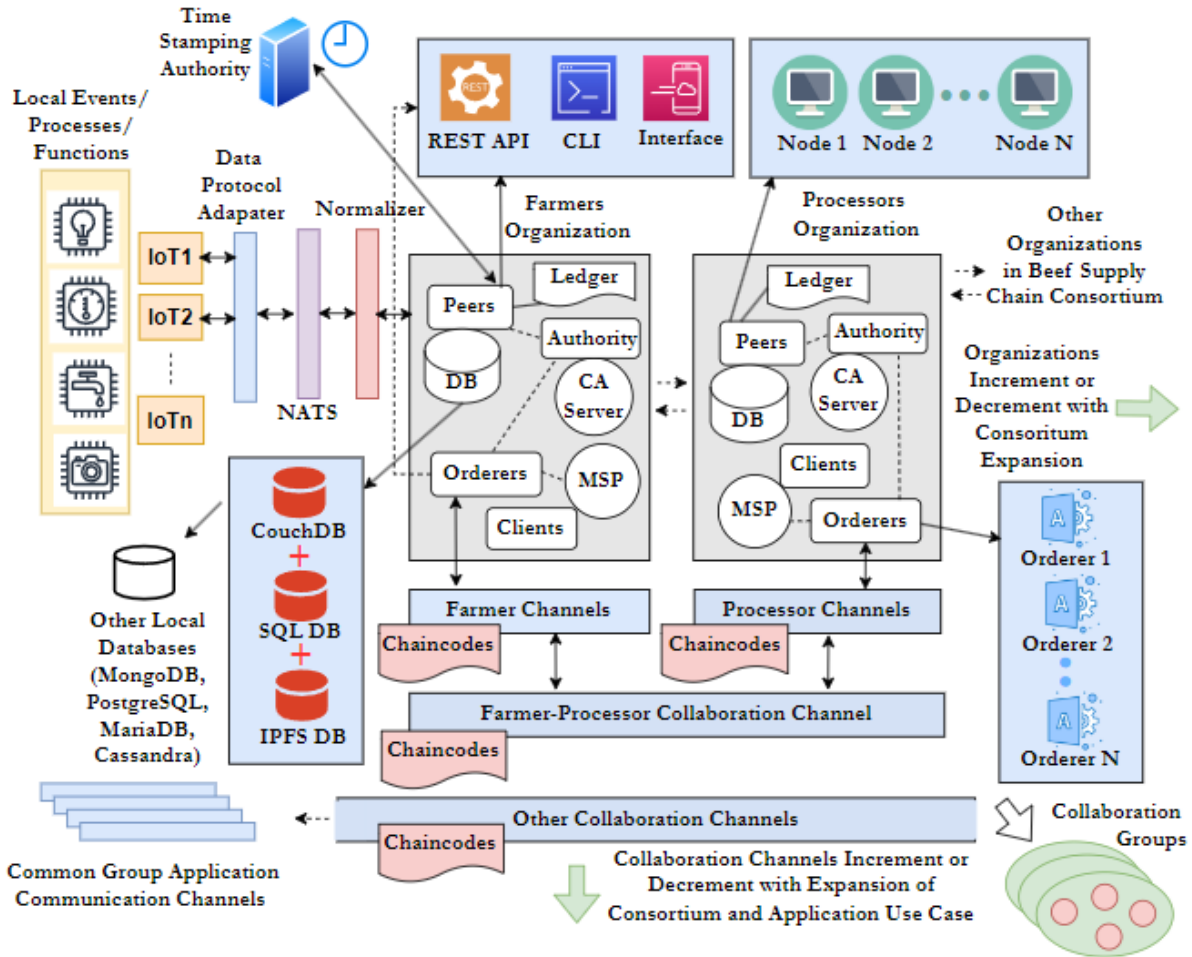


Figure 3.4 Distributed resources including Time Stamping Authority (TSA), databases and interfaces to consume data from internal and external processes are seamlessly integrated into the collaboration framework to enable applications such as traceability and common data (knowledge) sharing

directly storing data from supply chain events or for saving off-chain data. The distributed database (IPFS) runs as part of a shared data hosting service where IPFS nodes (peers) are spread across several organizations (as shown in Figure 3.6). This allows implementing distributed applications such as maintaining traceability records or enabling collaborative machine learning from federated data. Privately networked IPFS node clients can only directly view a file if they are allowed to view CID over shared blockchain channels. Both local (within organization) and distributed databases (spread across supply chain) also serve to record and maintain sizable off-chain data that cannot be conveniently stored in the blockchain. The framework extends further by integrating IoTs and other data generating sensor sources (with TCP/IP network interfaces) within each organizational domain

to allow extracting and storing data locally for processing and sharing. A Time Stamp Authority (TSA) service is integrated in the framework and available across the supply chain consortium to establish a time sequence of events, functions and transactions where data sequence cannot be established by blockchain. A TSA certifies the existence of a digital document at a specific time, ensuring its integrity and authenticity. This process is vital for legal validation and data integrity in digital transactions. This creates an added security layer for establishing timelines of off-chain events in relation to on-chain events stored in blockchain. A TSA running in a coordinated way serving multiple organizations by issuing standardized and consistent time stamps is referred to as a Universal Time Stamp Authority (UTSA) in the thesis.

3.5.2 Expanding collaboration groups into a fully decentralized framework

The proposed framework boots up from the collaboration initialization server from where organizations can form groups to manage (upload/download data) information required for setting up distributed yet connected applications. The framework can then extend to a single participant organization (e.g., farmer, regulator) or a combination of organizations (e.g., farmer, breeder and regulator) with communication channels to broker (share policies, decisions) and establish the purpose of collaboration. Once the purpose of collaboration (e.g., to record traceability data) is established, the group (networked consortium) continues to grow with incorporation of other vetted organizations. Each organization establishes and maintains their own resources (local databases and IoTs) in addition to connecting and maintaining distributed group resources (distributed databases and federated nodes). Participants can decide to form and join different collaborating group sets of organizations within the consortium with dedicated blockchain communication channels for managing supply chain applications. As long as there is at least one active communication channel (e.g., a beef-supply-chain channel connecting a group) running at one supply chain participant (e.g., administrator/maintainer), the network is kept alive and theoretically can expand to any size depending upon the type and number of applications involved. Group common applications could range from tracking inventory globally incorporating international regulatory authority to applications that involve collaboration between hundreds of distinct supply chains providing a

single item, e.g., a computer chip. Once a group starts from the collaboration server with the required resources to connect with other members, it can continue working or expanding without the group initiators need. Hence, the collaboration initiator server works as a network root and groups represent leaves. The leaves can get disconnected from root any time and can expand independently or form a new root (a vetted organization) to perform the same task of management of group resource information.

3.5.3 Collaboration groups functional layers and specific tasks

At the functional level, the proposed framework (as shown in Figure 3.5) is divided into two components namely the *Data* layer and the *Application* layer. At the *data* level, the framework is categorized into: (1) *control* layer, (2) *transaction* layer and (3) *storage* layer. At the *application* level the framework is categorized into: (1) *API* layer, (2) *extension* layer and (3) *protocol* layer. The *data* control layer is responsible for retrieval and transfer of data between network components (e.g., between data source and database storage) while maintaining data integrity. Data in the supply chain framework is either generated from sources like IoTs, is directly recorded into database storage from participants (clients) operating within organization tiers or is generated from processes, functions and events in the chain and recorded in database using pre-configured API routes. Regardless of the data source, all data for an application is generated and stored within the jurisdiction of a participating organization group or consortium with predetermined access (read, write, modify and delete) rights. Administrative rights are configured during the time of starting a service while generic user rights are configured when a user profile is created.

The *data* control layer is also responsible for restricting shared data within the jurisdiction of where it is authorized to be shared and used. The other part of the *data* control layer - the *transaction* layer - is responsible for moving data record from one participant jurisdiction to another, for example sharing traceability records of breeders with retailers so that the retailers can append their records while keeping the retailers information intact. The data transaction layer is also responsible for verifying, accepting and executing changes on the ledger database. For example, validating and changing ownership of supply chain assets. The data *storage* layer is responsible for securely

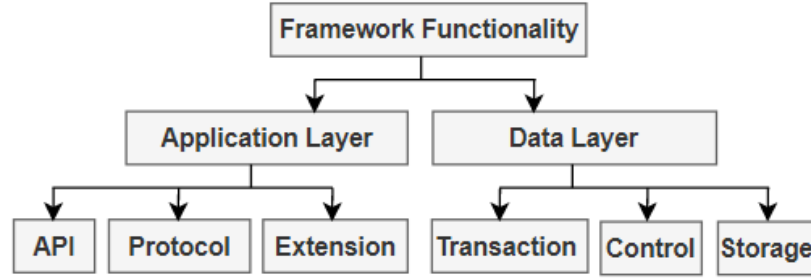


Figure 3.5 Overall functionality of a collaborating group is broadly divided into application layer and data layer

recording or updating blockchain ledger state in database, storing data records in database storage, indexing records and maintaining data reference pointers (for example CIDs of actual data in distributed databases).

To leverage heterogeneity and diversity of data originating from disconnected supply chain or organizational jurisdictions, we make use of distributed databases where segments of actual data (e.g., traceability data) are stored and maintained by participants that have vested interest in the data. To enable storing and handling complex data with diverse formats (*e.g., graphic, numeric, procedural pointers, object type, timestamps etc.*) generated in a complex disjointed supply chain like the beef chain, we make use of relational (PostgreSQL, MySQL), non-relational (CouchDB, MariaDB, LevelDB, MongoDB and Cassandra) and hybrid distributed database (IPFS). All organizations in our supply chain example (meat supply chain) make use of a mix of these types of databases. For example, animal records in farmer's organization are stored in relational databases because of the fixed nature of parameters (animal weight, color, age etc). In regulators domain, data is stored in non-relational databases because the regulatory files exist in diverse formats (e.g., graphical, numerical or Universal Character Set (UCS) format). Similarly, in regulators organization, data is stored in distributed databases where a number of regulatory groups come together to mutually maintain records (e.g., kosher and halal certifications).

The second functional layer of the proposed supply chain framework, the *application* layer, is responsible for allowing clients, computer programs (e.g., smart contracts) and resources (e.g., processing nodes), access to various functions (implementation of application use case) in the

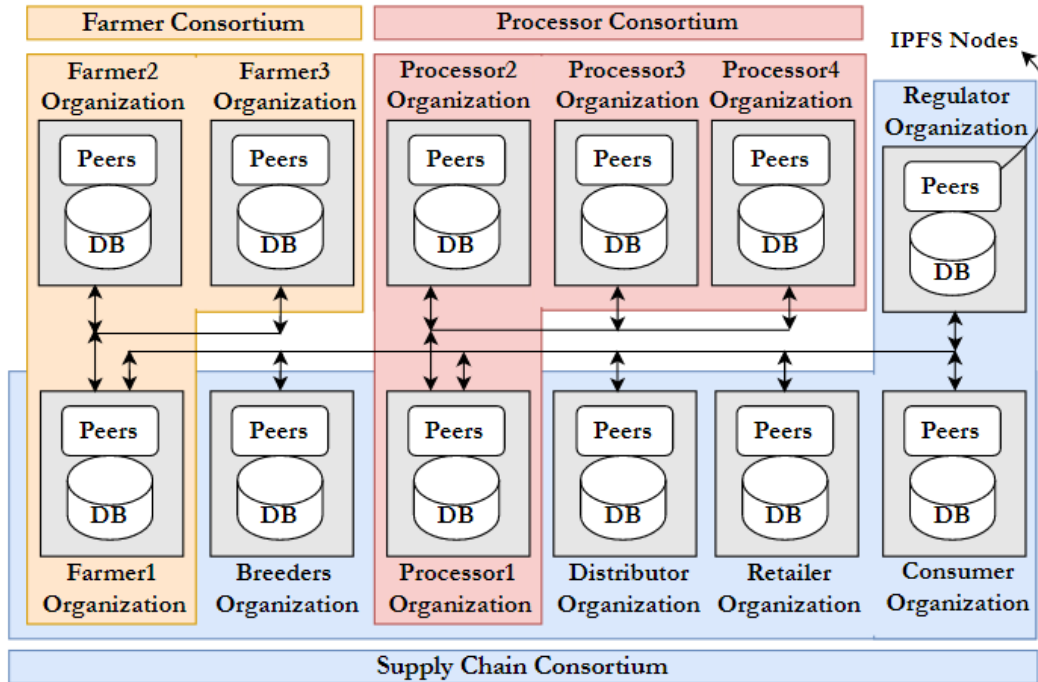


Figure 3.6 Distributed IPFS nodes in the supply chain are setup in different tiers of (privately connected) consortium of organizations (shown in different color shades) enabling collaborative data sharing and maintenance

framework. Particularly, the *API* layer provides clients, resources and other programs, a functional interface to execute various application use cases in the proposed framework. For example, an *API* in the form a *Linux* based Command Line Interface (CLI) has been provided in the framework for configuring the network (e.g., extending or trimming organizations or communication channels) to allow different organizational and consortium layouts to operate, communicate and share information. The API layer also provides clients or other computer programs (services) an interface to access various resources in the framework, e.g., an interface to read or write in databases, add or remove IoT devices, modify user rights, configure network settings or access communication channels. The second part of the *application* layer, the *extension* layer allows participants to either modify the already existing supply chain application functionalities or add more features to an existing application function. For example, adding a new organization in the consortium or integrating a new federated data node to an already existing configuration of connected nodes. Finally, the *Protocol* layer is responsible for seamless integration and communication between various components of the supply chain network with different functional implementations. An

example of this is where data generated from IoT devices interchangeably makes use of Message Queue Telemetry Transport (MQTT), Constrained Application Protocol (CoAP) or other formats to allow secure and consistent data type transfers, to reduce transmitted data packet size and to increase the frequency with which data can be reliably written to databases. Similarly, Hypertext Transfer Protocol Secure (HTTPS) and RESTful API protocols are used over Transmission Control Protocol/Internet Protocol (TCP/IP) web interface by clients or computer programs to securely access resources residing on various nodes in the network.

3.6 Conclusion

In this chapter, an overview of the supply chain collaboration framework was presented that facilitates connectivity for participants, allowing them to engage in various projects such as traceability and tracking. The system integrates multiple functional layers, including blockchain, distributed and local databases, and IoT sensors, ensuring seamless interaction while giving participants complete control over their respective components and data. This framework is designed to be both versatile and scalable, with applications that are driven by data. It incorporates stringent access control policies, a time-stamping authority, and a permissioned consortium infrastructure, all working in conjunction with blockchain technology to ensure a reliable and secure system. Collaborative groups are formed through the collective pooling and management of resources. Once these groups are established, they function in a fully decentralized and distributed manner, capable of scaling without the ongoing need for the original initiator. The framework's architecture not only supports dynamic collaboration but also guarantees data integrity, security, and participant autonomy, making it an adaptable solution for contemporary supply chain needs that arise from fragmented stages.

CHAPTER 4

IMPLEMENTATION DETAILS OF BEEFMESH FRAMEWORK

This chapter provides a comprehensive explanation of the implementation aspects of the BeefMesh collaboration framework. It begins by detailing the software tools and custom scripts utilized in the framework. Following this, it explores the blockchain layer and the distributed databases layer, explaining their roles and integration processes. Key topics such as data security, resource access control, and the management of authorizations for groups are thoroughly discussed. The chapter also covers the development of secure data pipelines designed for both local data consumption and global data sharing.

4.1 Software Tools and Custom Scripts

We implement the proposed framework by programming, implementing and integrating open source software tools that include IBM Hyperleger Fabric [141], GlusterFS [144], Docker [143], Databases (CouchDB, MongoDB, InfluxDB, PostgreSQL, MySQL, MariaDB, IPFS, Cassandra), Mainflux [145], Prometheus [146] and Grafana [147]. A starting point for any collaboration group in the framework is a consortium of organization (e.g., breeder and regulator) started using custom scripts [141] by pulling files and information from a RESTful frontend (group initiator server). The frontend is built using a Python flask application and the custom scripts are provided as generic Linux bash files. The front-end allows organizational clients to get registered, register a new collaboration group and manage (upload/download files and data) resources required to form a self

hosted distributed network. The frontend manages clients of a collaboration group by routing them to register and create active sessions before allowing them access to group resources. Among other resource information, some of the information required to form a collaboration group include: (1) Docker Swarm overlay manager and client keys (2) GlusterFS node addresses (3) IPFS private addresses (4) blockchain channel names (5) group, its purpose and the group members information.

4.2 Implementing the Blockchain Application Layer

The group initiator server along with the resource management tasks, starts a basic blockchain service consisting of an ‘administrator’ and ‘regulator’ organization. As groups are built they initially connect to the blockchain organization at the server with global blockchain channel (beef-chain-traceability channel) as shown in Figure 3.1. This allows a minimum setup of blockchain network alive for organizations to start building and connecting groups from scratch with expansion to any levels. Each extending organization starts a Certificate Authority (CA), Membership Service Provider (MSP) and an Orderer node service as a minimum requirement to support blockchain functions. The blockchain services use bridge and overlay networks to connect with other local databases including the distributed IPFS to allow managing local and distributed data storing and sharing. MSP nodes performs a number of functions for any group including setting up and maintaining root CAs, intermediate CAs, organizational units, administrators, revoking certificates, signing certificates, private keystores, Transport Layer Security (TLS) intermediate and root certificates. To minimize resource consumption (running services), all applications are implemented as lightweight docker containers with securely exposed and reachable TCP/IP network interface.

Each group maintains a single block channel alive (at the group initiator organization) as a starting point for other organizations to connect and expand from. This is particularly useful when the group started from the group initiator server but decides to continue on its own without the need from the server. The group starter organization can optionally also run the same service on their host that is run at the group initiator server to allow organizations to form and manage further groups and sub-groups. A single blockchain channel connecting organizations suffices the need for deciding policies, actions and collaboration goals while more channels can be helpful in running

isolated applications. Application channels exist and connect only the participants (organizations) that use the application. The decision to allow an organization to join an application channel is done collaboratively by the participants that are already part of the channel and sharing the decision and required resources through a group resource server (main server or other local group servers). Hence, application channels exist at different tiers supporting distinct applications, e.g., between organizations (inter-organization), within organizations (intra-organization), across the consortium (e.g., supply chain channel) or between consortium's (inter-consortium). An important aspect of the underlying implementation of the blockchain network for any group is that it is not disrupted and continue to work when organisations leave or join.

Blockchain application channels serve a number of purposes in our framework. Each channel supports more than one use case by utilizing programs (smart contracts) installed on each organizations peer nodes that are part of the channel and intend to use the functions that the program supports. We install programs (developed in GO language) on all organizational peer nodes with ability to store traceability and generic data in data structure place holders. The stored data can be an IPFS CID pointing to large data files or can be variables of different types (string, character, integer etc). The smart contract provides the functionality to store and retrieve data, check and change its ownership and restrict users that are not authorized to view or change data. By having different programs running on the same channel, a plethora of tasks can be supported. In our beef chain example, we make use of both intra-organization and inter-organization application channels. Consider a group of farmers where each group makes use of a common application channel to share animal dating information or data about their herd for sale with either all or a specific farmer group. To keep track of the chain of events on any application channel, all changes are stored in a *stateful* database (LevelDB) and the change itself (a transaction) on the channel is processed by a *orderer* node that validates the change by running a consensus (Byzantine Fault Tolerance (BFT)) algorithm [148], among the nodes that are part of the application. BFT is often expressed in terms of the maximum number of faulty nodes that a system can tolerate while still maintaining consensus: $f \leq \frac{N-1}{3}$ where f is the maximum number of Byzantine faults the system can tolerate and N is the

total number of nodes (including both loyal and faulty nodes) in the system. There can be more than one *ordering* node in each organization depending upon the groups within the organization and the level of reliability required from redundant nodes.

4.3 Managing Authorizations in a Collaboration Group

Application programs and resources in the framework are made accessible to clients (users) and nodes (server or application node) at different tiers (levels) by issuing certificates (e.g., X.509 security certificate) from CA and by creating working groups. Certificates issued by CA serve to enable privacy and security through TLS (TLS) based communication between different network elements (e.g., between a client and data server). CA also serves to register specialized nodes (e.g., databases, MSP and federated data nodes). Working groups for clients within each organization are created by enforcing roles through membership authorization certificates.

Another layer of reliability to track data manipulation is created by TSA server that is available to organizations through Secure Sockets Layer (SSL) communication for time stamping files using RFC 3161 Time Stamp Protocol (TSP) [149] (see Figure 4.1 and Figure 4.2). The Internet X.509 Public Key Infrastructure (PKI) TSP as defined in RF 3161 generates time stamps using hash and private keys. Let T represent the timestamp provided by the TSA, D denote the data (datum) that we want to associate with a specific time, $(H(D))$ denotes the hash value of the data (D), $(K_{\{TSA\}})$ denotes the TSA's private key and $(Sig_{\{TSA\}}(H(D)))$ represents the TSA's digital signature on the hash value of the data. Then the timestamp (T) generated by TSA can be represented as:

$$T = (H(D), Sig_{\{TSA\}}(H(D))) \quad (4.1)$$

For verification of timestamp, the hash value of received data is calculated ($H'(D)$) and matched with the original data ($H'(D) = H(D)$) after verifying the digital signature of TSA using a public key. Digital records from the TSA server are stored as Tracker Status Report (.TSR) in the blockchain which are used in conjunction with other records stored in database or blockchain ledger to establish provenance and traceability of data. Organizations using a common application (e.g., maintaining traceability records) use the same TSA server to verify and establish the time for modification of

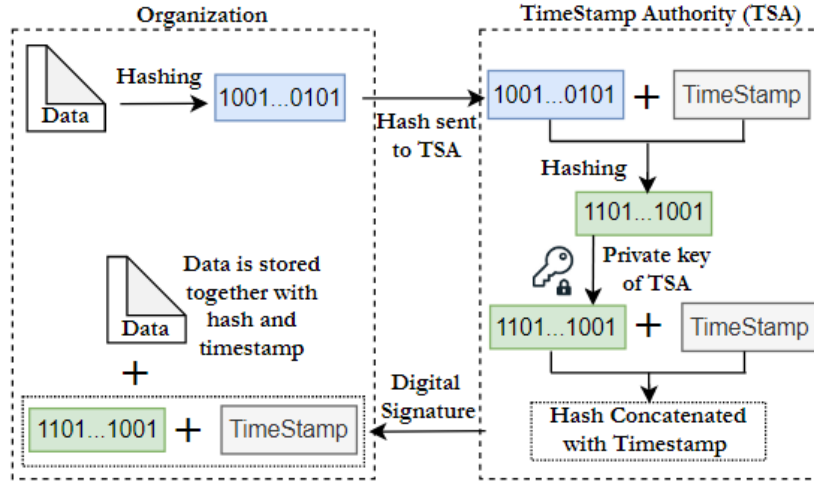


Figure 4.1 Sequence of time stamping for files of an organization using federated TSA

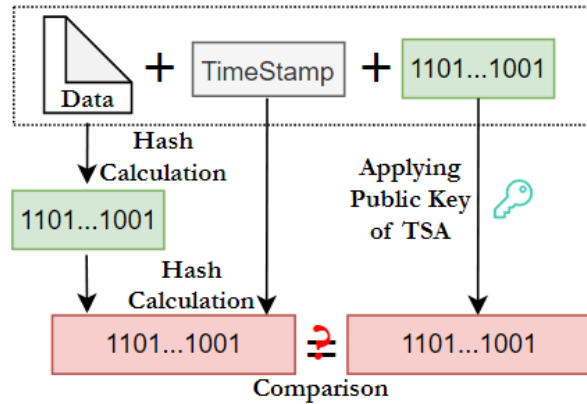


Figure 4.2 Checking correctness of timestamped files requires calculating hash function and utilizing public key of TSA node

files. The TSA server is integrated for new applications with existing organizations by initiating a request over a common blockchain application channel as shown in Figure 4.1.

4.4 Data Pipeline for Consuming Localized Data

Complex supply chains generate sizable frequent data from various events and processes at every step of the chain. To consume and store data generated at various steps of the chain, a number of data consumption interfaces are configured and connected to databases. The data consumption interfaces with messaging format conversion support (as shown in Figure 4.3) include: (1) Hypertext Transfer Protocol (HTTP) (2) MQTT (3) CoAP (4) Open Platform Communications- Unified Architecture (OPC-UA) and (5) Long Range (LoRa). These interfaces are used to adapt

and transfer messages to storage platform over secured channels without compromising the original characteristics of the data. For example, multimedia (sound, images, video) files from organizational devices are directly sent using HTTP messaging protocol over TCP/IP. Examples of such files could include video and audio recordings from animal farm or text files from gateway devices. MQTT is used to adapt and transfer messages from light weight queuing services that work on publish-subscribe mechanism (i.e. one machine interface publishes lightweight messages and some other machine is tuned to allow capturing the message over same channel). Examples of devices using MQTT include light weight IoTs used in gas sensors, lightning fixtures and refrigeration units. CoAP protocol is used for low powered devices working in constrained environments with low network bandwidth. Example of such devices include automation based monitoring tools for building and infrastructure such as energy utilization sensors. The OPC-UA protocol is used to adapt traffic from automation devices used in industrial infrastructure such as machine load or capacity monitoring devices. Finally, the LoRa protocol is used to adapt and direct traffic from long range yet low powered wireless sensor devices. The LoRa protocol is used with devices that detect and send abnormal or random readings from sensor devices such as the ones employed in monitoring air quality in agricultural fields or measuring soil parameters.

Since data from supply chain functions is generated with different formats (with different parameters such as frequency, size, criticality and sensitivity), we incorporate more than one type of database where the data can be stored for processing that follows the Extract, Load, Transform (ELT) or Extract, Transform, Load (ETL) based data engineering principles [150]. For storing large amounts of data transferred over HTTP we use MongoDB which is a document based Not only SQL (NoSQL) database using JavaScript Object Notation (JSON) type structure for managing data. For storing data from MQTT traffic, we use InfluxDB which is suitable for storing and analyzing large size time series data. For storing data from CoAP traffic, we use MongoDB and InfluxDB. For storing data following OPC-UA format, we use Cassandra which is a wide-column NoSQL database. Finally, for storing data transferred with the LoRa protocol, we use PostgreSQL which is ideal for storing relational data generated from organizational statistics,

for example wireless devices keeping track of type and quantity of supply chain stocks.

IoT devices and sensor devices play a vital role in gathering local data from organisations in a supply chain. Support for taking in data from IoT devices and sensors for utilization in our framework is provided by configuring and running addressable TCP/IP interface and connecting it with local databases. Software tools from open source Mainflux Labs application are used to implement a network of connected IoT devices and clients to manage the devices and direct input data [151] to a document-based database (MongoDB). MQTT and CoAP protocols are used for optimal, frequent and consistent data transfer between IoT devices and databases. A data-reader and data-writer service for IoT devices is provided in the form of a document-based database (MongoDB) and hosted as a docker container within our framework. Document-based databases allow a flexible data schema for IoT information to be stored and retrieved while allowing it to easily evolve with changes in data formats. Network Address Translation Services (NATS) sits in between the actual IoT devices and the databases to allow correctly routing data from numerous IoT sources into the organizational network which is then filtered with a *Normalizer* to convert data features to a homogeneous scale for wider application use. A *SenML* normalizer is used which provides a balance between the actual information and its source by optimally packing sensor information and actual data in each message.

4.5 Supporting Off-Chain and On-Chain Common Data

Storing sizable data on blockchain is considered inefficient and hinders scalability because of the fees involved with each data transaction and the time consumed in convergence of the underlying consensus algorithm. Hence, we solve the issue of recording sizable data generated frequently from events and processes at every step of the supply chain by extending blockchain storage with *off-chain* local and distributed databases. A distributed peer-to-peer storage in the form of a swarm of IPFS is used to store sizable data chunk as part of the blockchain network while eliminating timeout and processing overhead issues. IPFS is a peer-to-peer based distributed file hosting platform that ensures longer life, privacy and immutability of recorded information [140].

Distributed peer-to-peer databases help with improving network performance by increasing availability and data fault tolerance as data is split and rendered from more than one source. IPFS

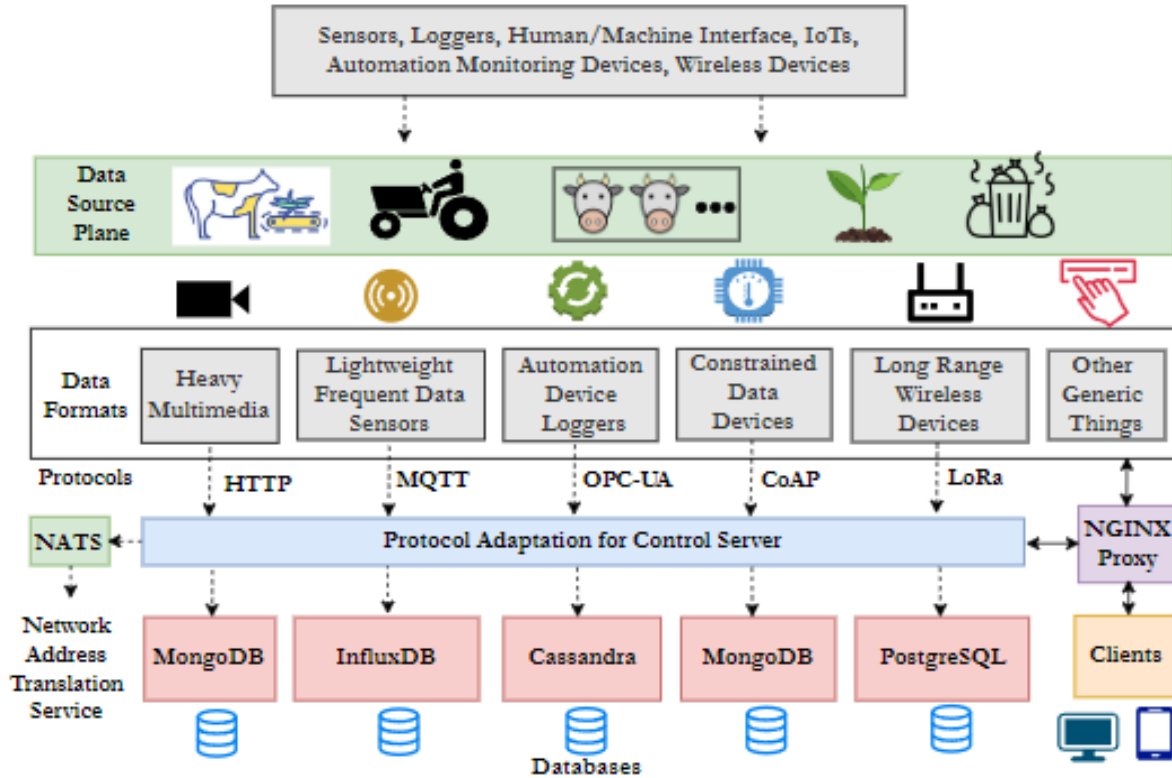


Figure 4.3 Data taken in from various types of devices, things and sensors in an organization is transmitted over different protocols using IP interface and stored in data format compatible databases

uses the concept of Distributed Hash Tables (DHTs) to manage metadata of information stored over peer-to-peer nodes. Examples of DHTs include the Kademlia DHT and Coral DSHT. IPFS also uses block exchange mechanisms similar to BitTorrent to coordinate between nodes in a network (called swarm), allowing chunks of information to be shared and stored. Another important IPFS feature is the version control mechanism which is similar to Git version control. The version control mechanism allows modeling and maintaining versions of files over time against major changes. IPFS uses Merkle DAG for version control because it helps recording changes to a tree of file system in a manageable way. IPFS also uses Self Certified Filesystem (SFS) which helps embedding certified server information of the IPFS network inside the address of the HostID of the remote filesystem. The HostID in the IPFS network and IPFS server location is therefore represented as:

$$\text{HostID} = \text{hash}(\text{publicKey} \parallel \text{serverLocation})$$

`/ipfsFileSystem/< serverLocation >:< HostID >`

where hash is a cryptographic hash function. Private or public nodes in the IPFS network are represented by two parameters which include a NodeID and a cryptographic hash of public key which is stored after encryption with a passphrase. In summary, IPFS is flexible enough to be used over any Transport protocol, can provision network level reliability with Micro Transport Protocol (uTP) or Stream Control Transmission Protocol (SCTP), provides connectivity using Network Address Translation (NAT) traversal method, provisions message integrity with hash checksums and authenticates messages using Hash based Message Authentication Code (HMAC).

By integrating distributed IPFS and blockchain nodes and connecting them in a decentralised framework, we implement a number of applications that leverage the supply chain organizational layout. For example, organizations append their distinct traceability data hash (or CID) to a common record over the blockchain ledger that keeps accumulating over time and can be monitored by a regulatory authority or shared with consumers. Distributed databases for consortium level applications are setup and networked in a manner where at least one database node resides in all organizations at a time. Distributed database nodes used for shared applications are connected in a private network setup in order to keep shared data available at all times (see Figure 3.6). Private IPFS nodes are also setup for applications where a group of participants within an organizations can collaboratively work on. Hence IPFS integration not only helps us with off-loading expensive data from the blockchain ledger but also helps with establishing and securely connecting different closed environment for participants to securely share data or collaboratively work on data applications. Finally, data from IoT and other sources that does not need to be stored on distributed or federated databases for sharing is offloaded to relational (e.g., PostgreSQL) and non-relational databases (e.g., CouchDB) available as connected containerized applications.

With sizable data generated and stored at each stage of the supply chain, data access restrictions are required for each user group in the organization (or consortium). Access to federated data

used for distributed applications in the framework is restricted by sharing CID (that points to actual data) that is available only to participants that are part of the application and the blockchain communication channel where data resides. The original data to be distributed over federated nodes is first serialized by the IPFS network along with its metadata and hashed using SHA2-256 algorithm. The hash is then converted to a CID that is stored in the blockchain ledger as a private asset property. For instances where more than one participant (group, organization or consortium) are using a shared smart contract program on a common application channel to keep federated distributed data available, only participants with access to CID can use (read/write/modify) the data. Access to CID is managed in the blockchain ledger by maintaining ownership groups for declared assets. For example, an asset could be an animal in breeder organization and as long as the animal is owned by the breeder, only allowed participants at different user groups within the breeder organization can make changes to the animals federated and distributed data. The ownership groups can be viewed as asset control zones used throughout the framework to limit read, write, modify, delete and data sharing operations by participants while ensuring data redundancy and reliability implemented through the use of multiple connected IPFS and blockchain nodes.

4.6 Maintaining Traceability with Data Splitting

A challenge in maintaining and establishing traceability data for events, functions and processes in complex supply chains comes from data splitting (also refereed to as data forking in the thesis). At certain points in the supply chain one traceable item gets split into thousands of components consequently resulting in thousands of data records with possibly different features pointing to the same source (item). Take the case of a processor (abattoir) facility in a beef supply chain where meat from an animal ends up in thousands of packages after processing. In some cases each processed beef package could be a mixture of meat from a number of animals.

To solve data forking issue and allow continuous tracking of actual data source over blockchain ledger, shared application channels are created that allow embedding CID in a number of steps (as shown in Figure 4.4). For example, CID of a particular animal asset over blockchain ledger can point to a list of CID each representing actual data for the processed meat package from the

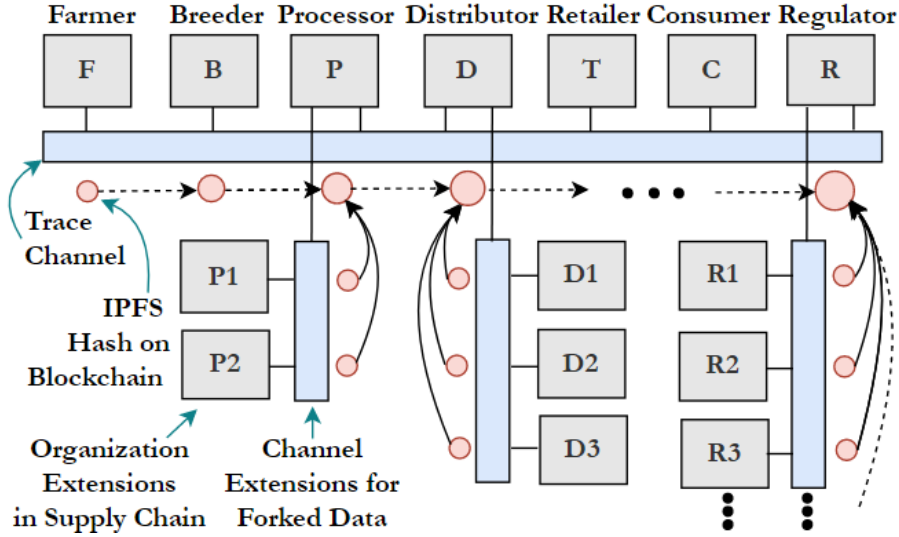


Figure 4.4 Split (forked) data in the framework is traced by aggregating layers of data CID in the main information tracking blockchain channel from forked or extended communication channels that connects different organizations

same animal but coming from different processors. Hence mixed data sources in a supply chain consortium can be easily tracked by embedding CID and storing it as a private asset property (metadata) over blockchain ledger as expressed below.

```

blockchain_metadata {

    ipfs_data_content_identifier

    ipfs_timestamp_content_identifier

    blockchain_transaction_context_identifier }

```

4.7 Knowledge Transfer Pipelines for Traceability and Collaboration

The proposed framework can be leveraged to enable a number of information learning, collaboration and knowledge transfer data pipeline architectures within organizational groups by using different resource configurations and integration of blockchain consortium. These learning architectures can be broadly categorized into: namely (1) Federated Learning (FL) and (2) Centralized Learning (CL) architectures. Here, we refer to FL and CL not only in terms of ML methods but also the way information or knowledge is transferred through the framework with a network perspective.

When considering FL from a ML viewpoint, it is an ML method where a particular algorithm is improved over time by individually training instances of it in a number of independent sessions utilizing parts of dataset residing in different databases [152]. Here we make a distinction between Federated Learning (FL) and Federated Database (FD) within our framework. FL is used in the ML scope and network perspective of information flow. FL involves iterative learning from databases owned independently by different clients whereas a FD is a system of databases where a particular dataset is spread across a number of different databases but is perceived as being in one database by the client. In FD, an application running on a server and serving a client request, transparently scans through all distributed databases making the client perceive as if the request was fetched from one database. In FD, a server keeps a consolidated record (catalog) of complete dataset and how its spread across multiple databases. Hence a FD can be considered a constituent of more than one database.

Use of FD in our framework is vital for supply chain participants mainly for two reasons, firstly cattle data is exceedingly large and keeps expanding overtime, secondly cattle data is managed by more than one entity, for example cattle at same breeder location can have data added by veterinary personnel as well as movement data integrated from field sensors. What makes our framework unique is the use of secure blockchain communication channels around FD and FL models in a collaboration group that allows tracking activities, files, progress overtime. Within proposed framework, FD is created for organizations by making connections to remote MYSQL servers where data resides and pulling required data into federated server thereby creating a unified view of data for the client (as shown in Figure 4.5). Parameters that are used in the connection string to pull data from remote servers include *server name*, *login credentials*, connection parameters (*address* and *port number*) and *table details*. For example, a general relational (SQL) table connection string to remote containerized server is given by:

schema://user[:pass]@host[:port]/db/table

where ‘schema’ is protocol for connection, ‘user’ and ‘pass’ are the login credentials, ‘host’ is the

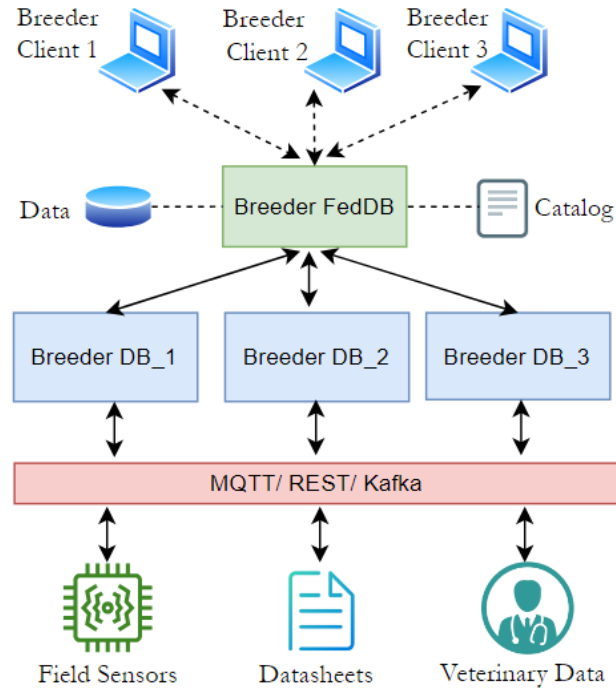


Figure 4.5 Collaborating organizations in proposed beef supply chain framework make use of Federated Database (FedDB) to provide a consolidated view to clients of common cattle data taken in from different servers

IP address of server, ‘db’ is the name of database and ‘table’ is the name of database table residing in the remote server.

4.8 Enabling Trust Through Hardened Framework Security

The proposed supply chain connectivity framework integrates a number of network components and resources that communicate over IP based interfaces (addresses, ports) using different types of networks (overlay, private, public). Critical resources and functions like user access, system nodes, databases, smart contracts, data flow pipelines, off-chain traceability data, federated data, containerized applications and ML pipelines (channels) are secured through a number of procedures and steps summarized below.

1. Providing secure login and authentication enabled with *Flask-Login* and *Flask-HTTPAuth* tools.
2. Securing containers with limited access to *OS* functions and *kernel* by running it as a non-root service from predefined *YAML* (docker-compose) format.

3. Using a continuous integration process which involves building up smaller container images that are sequentially packed together for deployment.
4. Carefully assigning privileges to containerized services with secured volumes that reside outside of the system directory and avoiding starting containers with elevated rights such as:
docker run --privileged -v /path:/path busybox -rf /path
5. Preventing containers from modifying routing or accessing *IPTables* by using *--capdrop ALL* function.
6. Sharing addresses, ports and network information for services with authorized users over secure channels.
7. Providing online and offline vetting for organizations to form or join groups by enforcing them to submit credentials for verification over collaboration channels.
8. Using alternate user group or *namespace* with less privileges than root *OS* for running containers.
9. With lower data ports being mostly reserved for server related accessibility, e.g., port 80 used for *HTTP protocol*, port 53 for *DNS* and 21 for *FTP protocol*, we bind the container related micro services with higher ports to avoid conflicts and security issues.
10. Providing *patches* for container updates when required and using most recent software tools.
11. A firewall *segmentation* is used between networked services to contain any breach and to secure interfaces.
12. We use minimal container distributions with no unnecessary binaries attached to it.
13. Accessing container services requires user credentials to be passed or accessed in the form of login passwords, security certificates, API tokens and encryption keys.

14. For user accounts to securely communicate internally and with outside applications, security certificates are provided through an authorized agent (server).
15. To secure database and blockchain ledgers in our framework, we create redundancy and regular backups so that total failure of services can be avoided.
16. A distributed peer-to-peer storage (a swarm of IPFS) cluster with swarm *keys* is used to harden security and avoid network breach.
17. To secure provenance of information, a universal time stamping server is used along with blockchain.
18. A GS1 code generation and compatibility checking program serves as a layer to reliably map traceability information to digital codes that can be tracked easily.
19. Sharing of machine learning model files (for example *.pkl*) is done over dedicated blockchain channels.

4.9 Conclusion

In this chapter, we discussed the implementation details of the proposed decentralized framework integrated with blockchain and distributed services to establish reliable and flexible information flow channels among participants. This framework connects supply chain participants without requiring modifications and can be seamlessly integrated with existing systems, making it a robust, non-intrusive tool for knowledge transfer. The novelty of the framework lies in creating a scalable and decentralized supply chain connectivity application that enables distributed participants to share vital information through connected collaboration groups. Participants retain full control over the collaboration network and information channels, allowing end-to-end storage, processing, and sharing of information across different organizational levels. The framework, with containerized distributed services and open-source secure integration interfaces, ensures easy connection with existing resources such as databases, sensors, and hardware, adhering to standard data privacy, security, and user control policies. By integrating distributed and local databases

securely connected to information channels, along with IoT and sensor interfaces, the framework supports diverse data sources from which information can be extracted. The proposed framework facilitates the extraction and secure propagation of critical information tailored for supply chain collaboration tasks using permissioned consortium hybrid blockchain frameworks and privately networked distributed database file systems. Controlled communication channels coordinate policies and decisions among group participants, such as jointly measuring and managing greenhouse gas emissions throughout the supply chain. The decentralized and distributed supply chain collaboration application incorporates solutions designed to protect user accessibility, control, data integrity, and confidentiality. It addresses software security measures and provides flexibility in adoption to foster user trust and transparency.

CHAPTER 5

TRACEABILITY EXAMPLE AND SYSTEM EVALUATION

In this chapter, we discuss a number of direct applications of the proposed collaboration framework, namely BeefMesh, with a particular focus on the traceability of cattle moving through the beef supply chain network. A traceability application is implemented and sample cattle data is used to track different domain specific parameters. Evaluation of different integrated services in the framework is done to analyze the efficacy of the proposed tool.

5.1 Traceability Example

To demonstrate the usefulness, value and efficiency of our proposed framework, we demonstrate a number of possible applications including traceability of cattle in the beef supply chain. The proposed framework initializes with a blockchain based traceability channel that serves as the starting point to initiate a network of collaborating organizations (as shown in Figure 3.1). Organizations within consortium then join the channel and start their own permissioned network resource with predefined configurations and communication channels. Each organization joins with different number of resources (e.g., peer and federated data nodes) depending upon the scenario. As a test case example for traceability, some of the organizations and resource shown in Figure 3.1, Figure 3.4 and Figure 3.6 are extended in collaboration group to include a breeder, 2 processors, 3 transporters, 4 retailers and 6 consumers. Since the system is distributed, participants can decide any time to join through new communication channels and group arbitrator for any application use and leave the groups without disrupting the network.

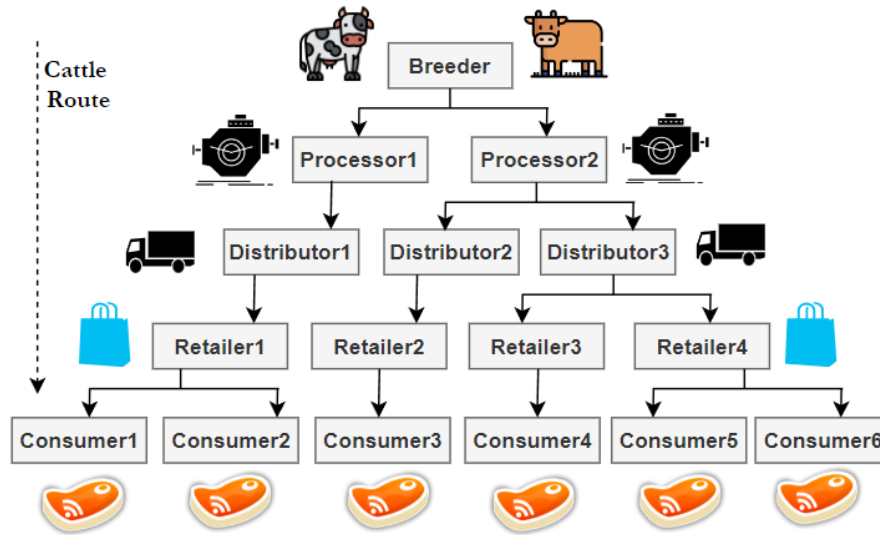
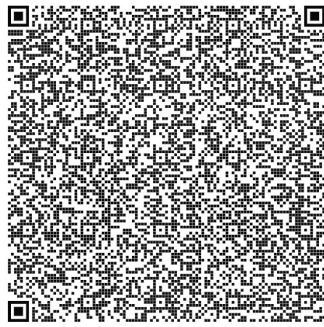


Figure 5.1 Starting from breeder organization, cattle take different routes through the supply chain

5.2 Data Logging for Traceability

Data is generated continuously (streaming sensor data) and in random manner (timed triggers) from different types of source residing in each organization within a consortium (see Figure 4.3). Static type of data is also generated at each organization for a cattle herd example case of 10 animals moving through the supply chain along different organization paths (see Figure 5.1). The static data parameters against each cattle generated at different organizations is shown in Table 5.1. Variable type of data generated from other sources can be differentiated based on its sensitivity, timing, frequency and size. Data sensitivity requirements mandate enforcement of privacy and protection rules. Data timing requirements mandate capturing exact time of event and the associated data for storing. Data frequency requirements mandate capturing data as soon as it is generated while accommodating a large number of data generation occurrences in a short interval. Data size requirements mandate availability of databases that can capture large number of parameters in relational or non-relational format. Overall, the different data types can be broadly classified into structured, unstructured and hybrid. Structured data such as sales, demand, financial, location and images are generated from Enterprise Resource Planning (ERP) systems, archives, sensors and scanning of barcodes/RFIDs. Unstructured data such as comments, internet clicks/hits, user preferences and texts are generated from internet or social media applications. Hybrid data



(a) QR Code

Meat Information:

Meat Type: Premium Beef Steak
Beef Rating: A+
Ingredients: Beef 500 gram
Fat Content: Less than 10%
Cold Chain: Yes
Consume Before: 31 Dec 2020
Storage: Maximum 5 Degree Centigrade

(b) Beef Type

Farm Information:

Cow Type: Angus
Certification: Organic
Age: 16 Months
Farm Name: Sparty Farm
Address: Sparty Farms, USA
Husbandry: Outdoor Grazing
Address: United States

(c) Farm Type

Transportation Information:

Cow Type: Organic
Veterinary Check: Yes
Date of Testing: 14 May 2020
Date of Transportation: 15 May 2020
Address: Sparty Transportation Services, USA

(d) Travel Info

Processing Information:

Cow Type: Organic Angus
Cold Chain: Yes
Address: Sparty Beef Processing Services
Certification: Organic
Date of Production: 12 May 2020
Date of Packing: 13 May 2020
Expiration Date: 31 Dec 2020
Lab Tested: Yes
Date of Testing: 13 May 2020

(e) Processing

Retailer Information:

Package Type: Premium Beef Steak
Type: Organic
Address: Sparty Beef Market
Expiration Date: 31 Dec 2020
Lav Verified: Yes
Date of Testing: 13 May 2020

(f) Retailer

Figure 5.2 Example of traceability data shown to a consumer using QR code. A single QR code pointer is linked to information generated at various stages of the beef supply chain network which gets fetched and unfolded at users ends

(structured and unstructured) such as quality and status reports of machinery are generated from different types of sensors. To consume and process information from a wide range of these data sources, different databases are started locally in the collaboration group and integrated into each organizations network. Local databases include MongoDB, MariaDB, InfluxDB, PostgreSQL and CouchDB as shown in Figure 4.3.

5.2.1 Configuring traceability channels

For recording regulatory processes and tracing publicly available events that cattle undergo, singleton or aggregated data values (depending upon the data type) over a certain time period are sent to a federated regulatory authority (e.g., NGO in our case) over secure blockchain communication channel. The regulatory authority keeps a federated record of the data agreed upon by the participating organization. Traceability data parameters are decided in three ways. First, one or more than one of the participating organizations can propose a set of parameters which are finalized

Data Type	Example Parameters
Veterinary Data	LabID, SamplingDate, ReportDate, FarmID, ZipCode, SamplerID, TestResult, AnimalVaccinationInfo, MethodType
Farm Data	FeedDeliveryDate, FatteningAnimalCount, AvgDailyGain, FeederType, AnimalArrivalDate, FarmsRegisteredName, AnimalDateOfBirth, HumidityPercentage, WeaningDate, MaxTemperature, MinTemperature, FarmsQualityCertificate, WeaningWeight, LotNumber, FeedQuantity, TemperatureValue, FeedType, FeedReceipt, AnimalDietPlan
Breeder Data	HeiferID, SteerID, AnimalID, ParityRate, InseminationDate, AbortionRate, RegistrationID, GeneticMakeup, AnimalOutDate, AnimalPhenotypeInfo, AnimalFinishingWeight, EstimatedBreedingValue, AnimalColor, BreederID, InseminationType, CalvingDate, DeadAnimalCount, AnimalBirthWeight, AnimalTraits, AnimalDam, AnimalSex
Processor Data	RumpData, IntraMuscularData, RibEyeData, BackFatData, HotWeight, ColdWeight, HarvestingData, AnimalID, MeatMass, FatScore, SheathScore, FrameScore, HipData, MarblingScore, CarcassClassification, CondemnationData, CondemnationReason, BeefGrade, NavelData, UltrasoundWeight
Distributor Data	DateOfDeparture, DateOfNotification, DepartureFarmID, ArrivalFarmID, DateOfDelivery, BillOfLanding, ExportDocument, BeefArrivalDate, HaulersID, DriverID, Consignment Detail, ConsignmentCondition, AnimalCount, ColdStorageData
Customs Data	ContainerID, ContainerClearanceStatus, ConsignmentDetails, SignOffDetails, ImportDocument, ExportDocument
Retailer Data	DateOfDelivery, SignOffDetails, BeefCutType, BeefProductCount, BeefProductPrice, TotalProductCount
Consumer Data	AnimalHistory, AnimalLifeCycleDetail, OptionsData, ConsumerRating, HalalCertification, KosherCertification, OtherCertification
Regulatory Data	AuditingData, SafetyCertificateIssued, FarmQualityCertificateIssued, OtherComplianceData

Table 5.1 Data parameters originating from different organizations used in the proposed beef chain collaboration framework

through voting. Second, regulatory authority can decide the parameters among which a voting method determines the ones to be used. Third, regulatory authority can work with an NGO or government authority to come up with the required parameters that can provide an average estimate of publicly visible traceability for all participants after voting.

Resource utilization (e.g., electricity and fuel consumption) and traceability data (e.g., cattle immunization) from organizations is stored in IPFS database and a reference (CID) of it is stored in the blockchain channel (see Figure 4.4) from which regulatory authority can download and unfold different parameters for users/clients when required using a Quick Response QR code. For data that is not shared with the regulatory authority, the supply chain participant keep it federated locally for ML applications and only share non sensitive insights with others. Through collaborative voting, the regulatory (NGO) node downloads resources (e.g., emission parameters) that need to be run against any participant data that they shared, for example calculation of standard greenhouse gas emissions against organizations resource utilization. A relational database is used to store static traceability data at the regulatory authority maintaining it. For test case, we use an example of 10 animals consuming resources and moving along the chain on different routes. Between each organization, transportation also takes place and therefore fuel is consumed in addition to usage of processes like cold storage during distribution. The route followed by the example cattle from farm to fork through the supply chain is shown in Figure 5.1. At the end of the cattle journey, publicly available traceability data (stored at NGO node and pulled and rendered by consumer node) is shown as in Figure 5.2.

5.2.2 Managing data sparsity and data size

In order to provision traceability data, there needs to be a global perspective of the vital information stored at different sub-organization levels. For example, the identification numbers of cattle should be in a form such that they can be easily mapped to the identification numbers on the beef packages after harvesting. Different sub-organizations in the beef supply chain network are independently owned. For example farmers raise cattle independently from the breeder requirements. Processors are monitored and controlled by independent private companies that supply beef prod-

ucts to different distributors according to their financial and logistic requirements. Our proposed connected framework not only allows collaboration groups to process and convert data according to each participants global or local view of beef supply chain, but also helps to provide summarized information of underlying events to optimize processes at sub-chain level. The majority of users at different levels of the beef supply chain, for example at the breeders level, cannot clean and process data before recording it on the ledger. Most often this results in the same type of information recorded in different formats, for example identification numbers of cattle may exist as ‘enrollment IDs’ or ‘tag numbers’ in records in addition to existence of invalid data, for example ‘0’ or ‘NAN’ type data.

Through a collaboration group recording sequence of events on blockchain and distributed databases, participants can extract, clean, process and cross match the data in an efficient searchable form whenever required. Different sources of data and the types of records they could produce in a beef supply chain network and used in our framework for cattle is given in Table 5.1 (some parameters adopted from [56] and others added with consultation from domain experts). We employ industry-standard GS1 codes to effectively record and transform triggered events into corresponding traceable unit metadata. This approach ensures a high level of traceability and reliability throughout the data capture process. The versatility of our system allows for the seamless capture and dissemination of a wide array of data types. This includes information from laboratories conducting various tests, reproduction facilities managing breeding programs, feeding houses monitoring animal nutrition, and logistics operations tracking the movement of goods. Additionally, data from meat quality certification agencies, carcass examiners ensuring compliance with standards, and weather reporting stations providing crucial environmental data can easily be integrated into our network. By enabling the capture and sharing of diverse range of information, our system enhances transparency, efficiency, and coordination across the entire beef supply chain. This comprehensive data integration supports informed decision-making and fosters collaboration among all stakeholders involved in the beef production and distribution process.

5.3 Testing System Performance

Once the system is up and running and a beef mesh application is laid out in a decentralized manner comprising multiple organizations from breeder, processors, distributor, retailer and consumers, different system tests are performed. First, a number of calls are made to timestamp files for storing on blockchain channels. Files of different sizes and formats are used (as shown in Table 5.2). Files of larger size take more time (as shown in Table 5.2) but the time to verify files average around 7 seconds (as shown in Table 5.3) because it only required verifying smaller timestamped files. Files in increasing size order (from 1MB to 200MB) from Table 5.2 are timestamped by making consecutive calls to times stamping authority running over overlay network and accessible to all organisations. For each file, consecutive calls are made for 10 seconds followed by next file in the order of size. System parameters are measured and reported in Figure 5.3 which include (a) received throughput (b) transmitted throughput (c) CPU usage and (d) memory usage of UTSA container running at main server. The collaboration framework was setup by organizational domains located in the Eastern Daylight Time (EDT) zones, hence timeline (x-axis) shows the exact time in (UTC-4:00) when the organizations were operational and experiments were run. On average, received throughput remains capped at 800 B/s while transmitted throughput remains capped at 6 KB/s making the timestamp functionality ideal for running against most of the critical files. The time stamping server does not burden the main server as it consumes only 1.76MiB of constant memory and CPU usage does not go beyond 0.23% for peak activities as shown in Figure 5.3(c).

Next, we test different databases running locally in conjunction with blockchain and IPFS nodes. Time taken to record different files (ranging from 1KB to 73MB) were tested to see how it could affect the overall system performance. Almost all of the files take more time to store when the size increase but it also depends on the type of the file (as shown in Table 5.6). JSON files took less time to upload specially on a diverse format supporting databases such as MongoDB while Cassandra took more time to store cattle records in the form of CQL files. As shown in Figure 5.6(b), Cassandra also takes a lot of resources because of running clustered database. Though databases such as Cassandra would be ideal for storing complex files in a beef chain network, it is

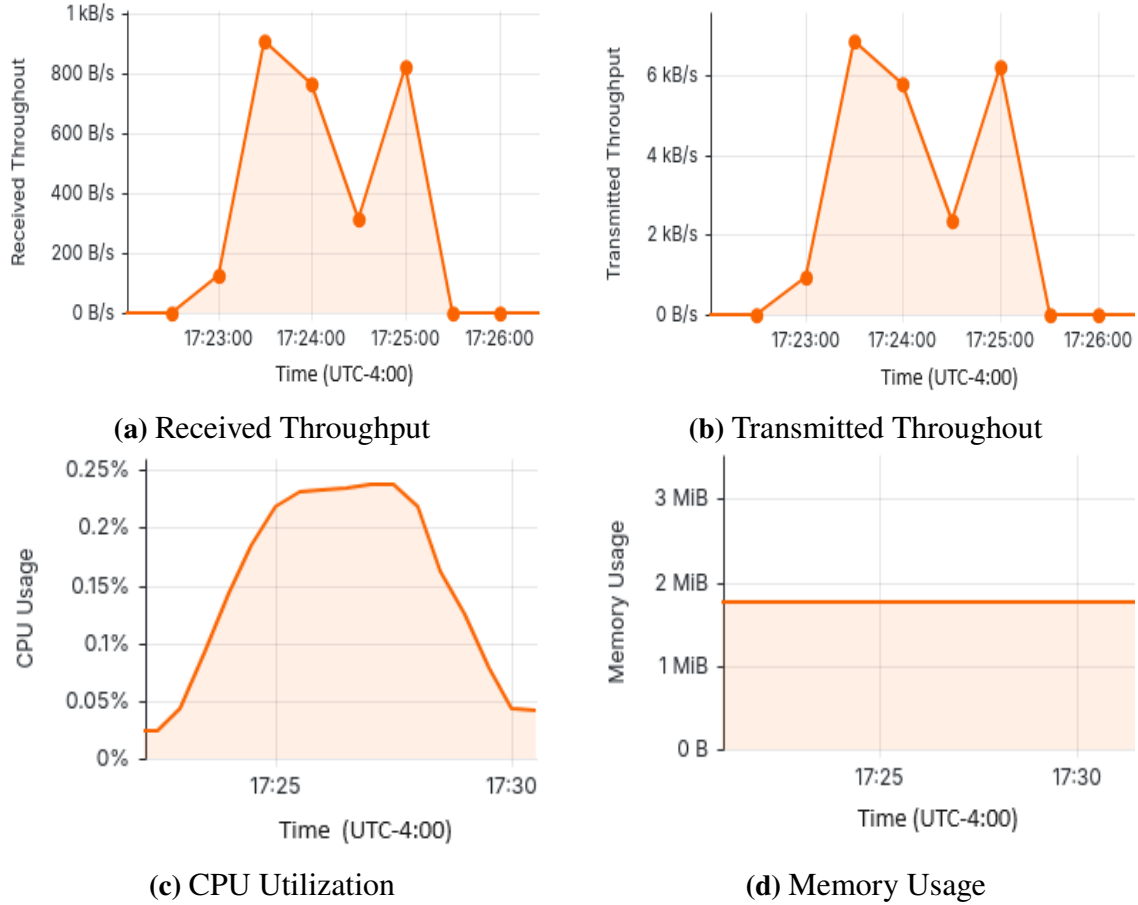


Figure 5.3 Files in increasing size order (from 1MB to 200MB) from Table 5.2 are stamped by making consecutive calls to times stamping authority running over overlay network. For each file, consecutive calls are made for 10 seconds followed by next file in the order. System parameters measured include (a) received throughput (b) transmitted throughput (c) CPU usage and (d) memory usage of UTSA container

Type	MD	JSON	PDF	MP4	SQL	JPG	ZIP	TXT
Size (MB)	1	5	10	15	24	50	100	200
Time (ms)	69	40	51	92	98	221	251	477

Table 5.2 Time taken to timestamp files of different sizes and types from time stamping authority

not ideal to be run on machines with less memory and storage. Hence, in most circumstances, a mix of MongoDB and PostgreSQL databases would suffice for most supply chain participants as heavy databases like Cassandra can consume a lot of CPU power for recording nominal sized files as shown in Figure 5.6(a). On average, MariaDB takes memory of 128MiB, MongoDB takes memory of 166Mib, PotgreSQL takes memory of 168MiB and Cassandra takes memory of 2.57GiB at startup on the host nodes.

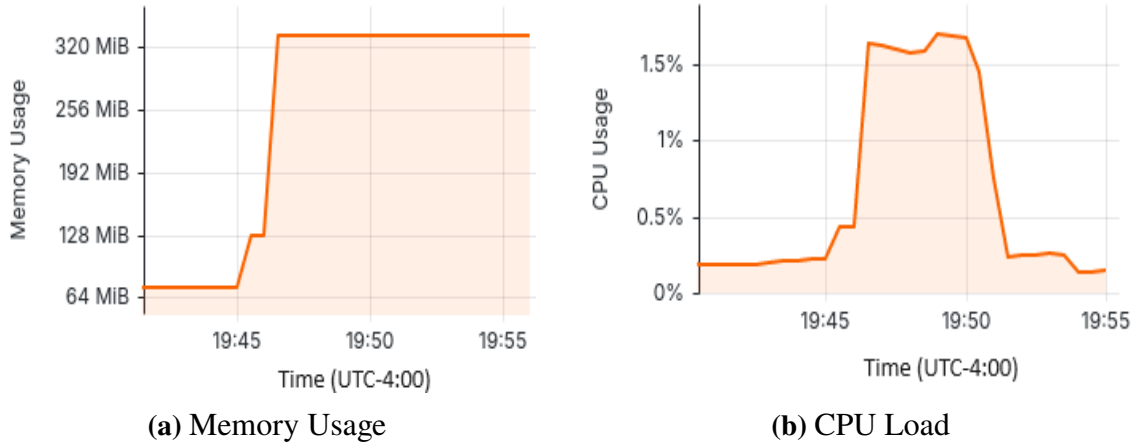


Figure 5.4 Collaboration server's memory usage directly depends on the size of content (files) uploaded for a group with around 1% increase in CPU usage during heavy tasks

Next, we test IPFS network working in combination with blockchain network storage. Files of different sizes (ranging from 1MB to 200MB) were uploaded to nodes connected in a private network setting. We upload the files on one organisation's IPFS node and download files from another organization's IPFS node. Files took different times to upload depending upon the size of the file as shown in Table 5.5. Zip files took more time to upload because the hash of the content is first calculated and files embedded in some form would require extra effort for decompression. Almost all types and formats of files get uploaded in very short period of time as highlighted by Table 5.5. IPFS in combination with blockchain, therefore provides a very powerful feature for managing supply chain data in a decentralized manner. It was also noted that once files are uploaded, it takes a very small amount of data exchange between IPFS nodes for a couple of minutes to sync information required for the files. As shown in Figure 5.8, all of the privately connected nodes took a couple of minutes to sync metadata information related to files that are uploaded, with maximum data packet exchange size of around 60B/s. The IPFS nodes memory usage directly depends on the size of content stored on it and increases proportionally as shown in Figure 5.5(b). During the file processing time, CPU usage of IPFS container node increases slightly to around 0.1% as shown in Figure 5.5(a). The initial memory consumed when IPFS node begins functioning is around 16MiB on average making it ideal to run at participating organization level.

Local IoT nodes and services were also tested to see how well the application is suited to

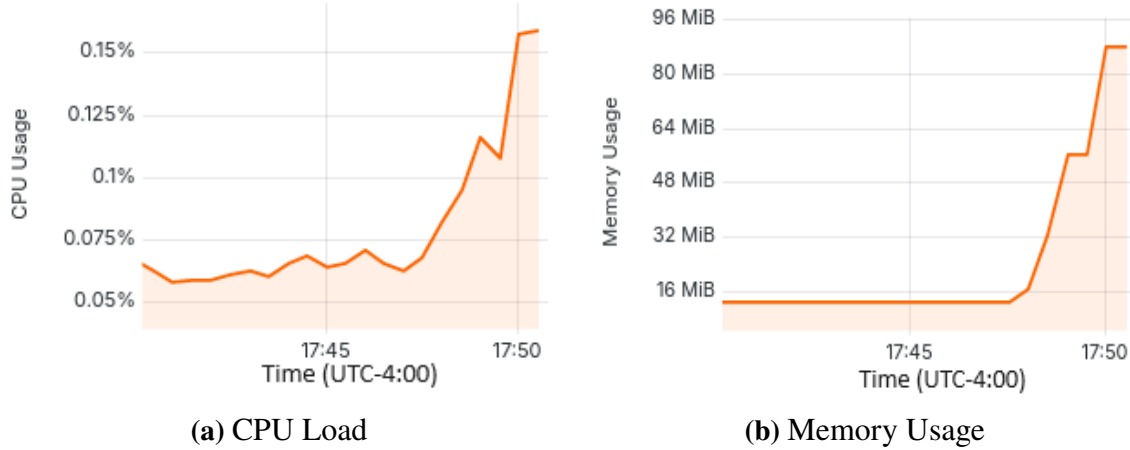


Figure 5.5 IPFS container nodes memory usage directly depends on the size of the content uploaded with a minimal increase in CPU utilization during file processing.

be run in combination with blockchain and IPFS nodes. IoT related containers (with around 15 sub-services) start with a load of 512MiB and CPU utilization of 2% with increase to 1.25GiB and 4% as all services coordinate together to maintain sensor data (as shown in Figure 5.7). This makes it suitable to be run on normal machines as long as there is enough back up memory to consumer fast sensor data.

Next, we test collaboration servers load by performing a number of tasks as shown in Table 5.4. Main tasks that were run included registering groups and users, logging in and out of server in addition to running multiple tasks against each groups resources. Registering users and groups was done in a short period of time (normally less than 50ms) even when consecutive calls were made from different organizations during the same time period highlighting its ability to cater for multiple supply chain groups during heavy loads. Text files with lots of user group data (200MB) took considerable longer time to extract and upload to databases compared to smaller visual (JPG) or data collections (text files). The download times were faster than upload times as shown in Table 5.4 which could also depend upon the type of network connectivity and data rates available to overlay network. The collaboration servers memory usage directly depends on the amount of data stored for groups and increases linearly as shown in Figure 5.4(a). Consuming larges files also comes with increased CPU usage as shown in Figure 5.4(b). The starting memory when collaboration server begins functioning is around 64MiB on average making it ideal to run at

Type	MD	JSON	PDF	MP4	SQL	JPG	ZIP	TXT
Size (MB)	1	5	10	15	24	50	100	200
Time (ms)	7	12	7	6	7	6	8	7

Table 5.3 Time taken to verify timestamped files from time stamping authority

Task Performed	Time Taken (ms)
Register User	46
Login User	52
Logout User	57
Create Group	52
Text File Upload (11 bytes)	49
JPG File Upload (50 MB)	791
TXT File Upload (200 MB)	6567
TXT File Download (200 MB)	637

Table 5.4 Collaboration server task performance efficiency

File (MB)	1	5	10	15	24	50	100	200
File Format	MD	JSON	PDF	MP4	SQL	JPG	ZIP	TXT
Upload (ms)	208	290	371	395	484	827	1391	1994
Download (ms)	229	401	449	635	743	1646	2747	2143

Table 5.5 IPFS upload and download times for various files

organization level or for self-hosting group management tasks.

We further performed a number of tests to determine blockchain infrastructure's task performance capabilities as shown in Table 5.7. Main tasks that were run, included initiation of

Database	File Type	File Data Size	Time Taken
MariaDB	SQL	1 KB	129 ms
		25 MB	3686 ms
		73 MB	9592 ms
MongoDB	JSON	1 KB	156 ms
		20 MB	1579 ms
		34 MB	1700 ms
PostgreSQL	SQL	1.5 KB	151 ms
		20 MB	1293 ms
		50 MB	6415 ms
Cassandra	CQL	1 KB	387 ms
		20 MB	2531 ms
		50 MB	8563 ms

Table 5.6 Time taken to write data of various sizes to different databases

Task Performed	Time Taken (ms)	Comment
Start Organization	19917	Two Orgs with two CA's and a CLI container
Register Clients with CA	4056	Registering and enrolling admin, peer and client
Register New Peer	13309	Utilizing existing Org's CA
Remove A Peer	3942	Updating channel and removing volumes
Register New Orderer	24915	Using new CA and service
Create New Channel	10357	Installing on two existing Orgs
Deploy Chaincode to Channel	53988	Installing on two Orgs with install, approve, commit and invoke lifecycle
Update Parameters on Chaincode	122	Writing a IPFS CID to an Orgs traceability record
Retrieve Data from Chaincode	101	Retrieving IPFS CID record

Table 5.7 Blockchain infrastructure task performance efficiency (Org refers to Organization)

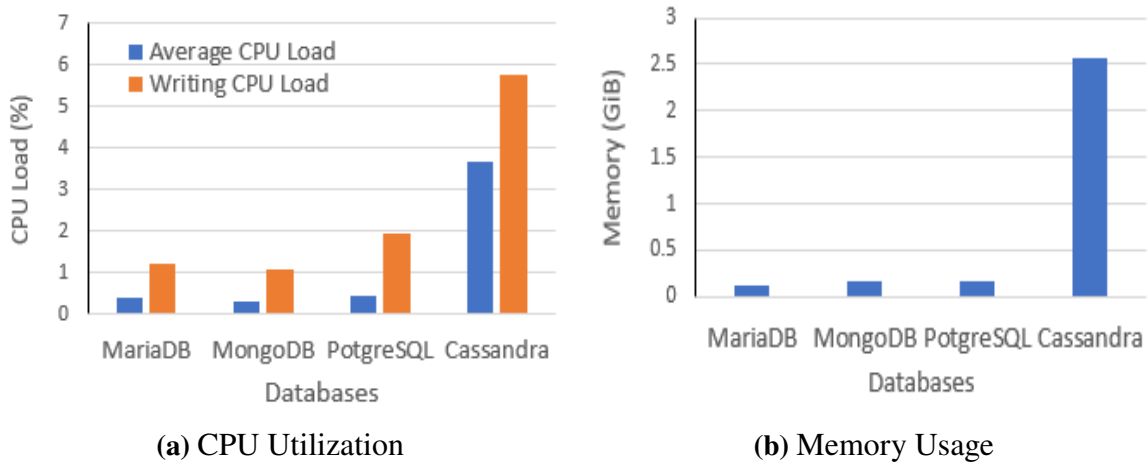


Figure 5.6 CPU and memory usage of local databases

new organizations, expansion of existing blockchain services, creating new channels, installing chaincode and, writing and reading data from blockchain channels. Starting a set of two new organizations (e.g. manager and regulator) from scratch took less than 20 sec which included starting CA containers, registering set of admins, peer nodes, client users, initiating and joining a channel, starting volumes and registering the services and volumes with a CLI container. As both hosts started on the same host, this gives us the baseline around which services running on multiple hosts can be compared. Similarly, all other blockchain tasks took less than 20 sec except for the creation of a new Orderer node and chaincode deployment on a channel utilized by two organizations. Initiating a new Orderer service takes more time than other task because it is usually a good practice to register it with a new CA. Further more, Orderer can take up some time to sync

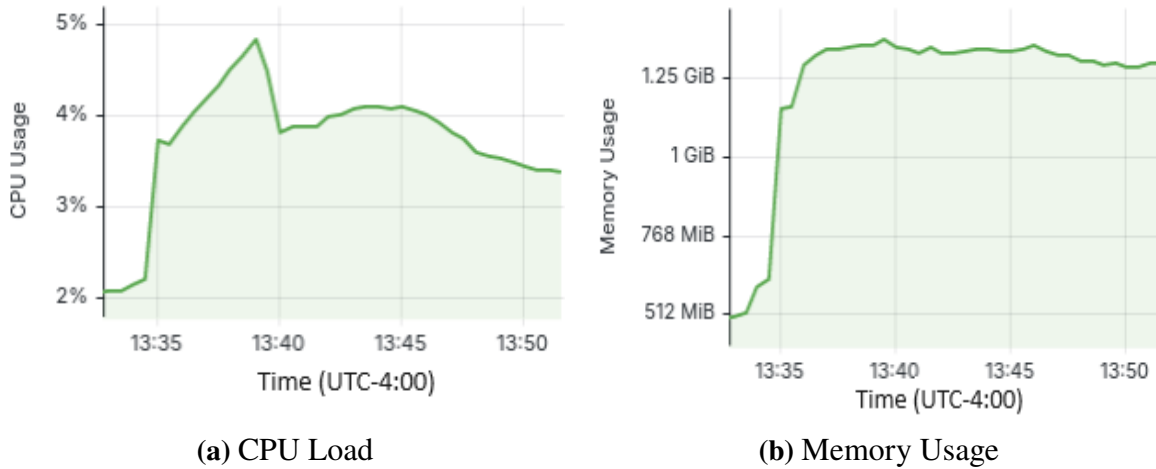
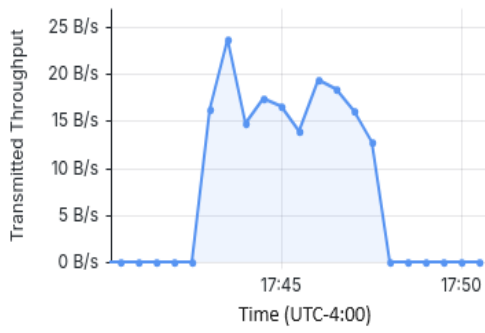
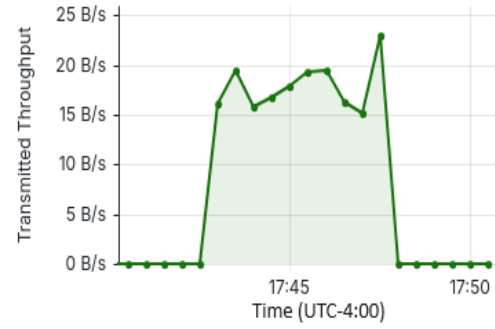


Figure 5.7 CPU and memory usage of IoT nodes and sensor application

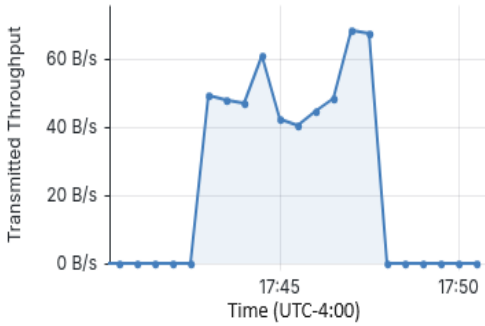
with channel updates. The longer a channel is up and running, more time the Orderer takes to catch up with the changes in blockchain state. Nevertheless, it took less than 15 sec for Orderer to join the channel through channel update procedures. Lastly, it took around 54 sec for chaincode to be installed on a channel that is shared with two organizations. The program size was roughly 15MB and the chaincode was installed following the install, approve, commit and invoke chaincode lifecycle for both organizations. Once installed, it took around 0.1 sec to read or write data from the installed program. The blockchain infrastructure task performance results provide encouraging results considering that it was run on a VM with mediocre capabilities and that a number of services would be required and can still easily be supported alongside blockchain to enable collaborative tasks. This is also validated by the combined average CPU usage and memory usage of the two peer node services running at the host node (as shown in Figure 5.9). The average CPU usage was around 0.651% and the average memory usage was around 35.51MiB for both of the peer services combined, with small CID related read and write tasks performed over the duration of the observation. It should be noted that the results were achieved assuming that the blockchain clients can directly access Orderer services on other nodes. When an Organization has not started its own Orderer service, a small amount of time will be added to push channel changes by exchanging channel update files through collaborator server.



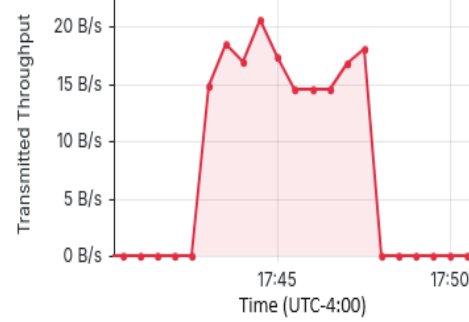
(a) Farmer Node



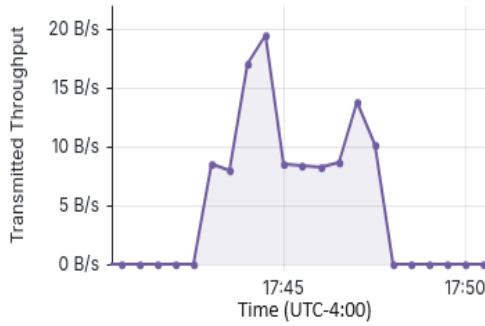
(b) Breeder Node



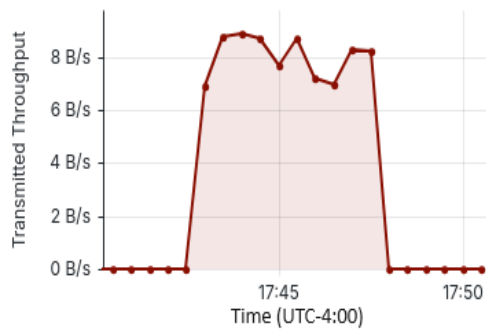
(c) Processor Node



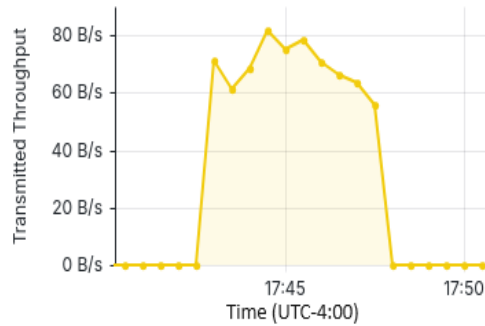
(d) Distributor Node



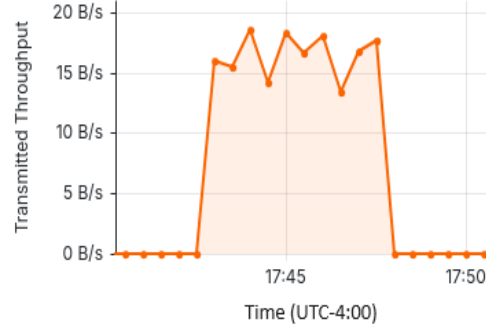
(e) Retailer Node



(f) Regulator Node



(g) Consumer Node



(h) Manager Node

Figure 5.8 IPFS data nodes syncing metadata after files uploaded (1MB to 200MB)

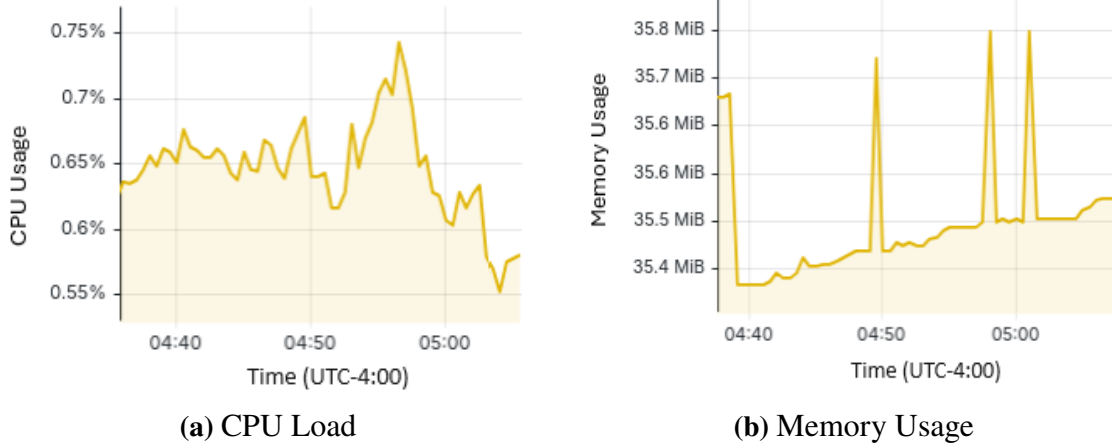


Figure 5.9 Average memory and CPU usage of the blockchain peer service nodes

5.4 Direct Traceability Applications

We use blockchain in combination with IPFS and other types of structured and non-structured databases to allow storing records for a number of other applications. Some of these include storing records for: (1) certifications (2) cattle movement (3) critical processes (4) critical events (5) resource usage (6) snapshot of data (7) snapshot of inventory (8) Information of groups using BeefChain (9) ID's and addresses of devices and nodes using BeefMesh. With a carefully designed integrated digitized beef supply chain system with privacy preserving information sharing, in addition to overcoming the current beef supply chain limitations, it is also possible to develop applications for optimizing system functions. Some important applications could include the option to remove any forms material scarcity, uncontrolled price hikes, unavailability of freights, traffic congestion while forecasting market demands, incorporating consumer response and allowing transparency [153].

5.5 Conclusion

In this chapter, we explored the direct applications of the proposed collaboration framework, focusing on the traceability of cattle within the beef supply chain. By implementing a traceability framework and utilizing sample cattle data, we tracked various domain-specific parameters. Through evaluating the different integrated applications and services, we demonstrated the efficacy of the proposed tool. The BeefMesh application example highlighted the system's effectiveness in

enhancing connectivity, collaboration, policy sharing, traceability, knowledge transfer, and value for participants. Our evaluation criteria including flexibility, scalability, reconfiguration, security, privacy, and data cost efficiency confirmed the robustness and utility of the framework in real-world scenarios.

CHAPTER 6

TRACKING CARBON FOOTPRINT USING BEEFMESH FRAMEWORK

Note: The contents of this chapter, either in part or in full, are under submission to a conference. The authors of the manuscript in the order listed in the actual draft include Salman Ali, Cedric Gondro, Qiben Yan and Wolfgang Banzhaf.

The beef supply chain significantly impacts the environment through various activities occurring in different stages. A primary challenge in mitigating these impacts is the difficulty in tracking the carbon footprint due to the lack of vertical integration across various stages. To address this, in this chapter we utilize our proposed blockchain-based collaboration framework integrated with IoTs and databases to capture detailed emissions data throughout the supply chain. In particular, we extend the BeefMesh application to allow precise carbon emissions tracking. The application further ensures privacy, transparency, and facilitates reliable traceability and scalable environmental data sharing, ultimately promoting emissions reduction and sustainable practices for the beef industry.

6.1 Basics of Carbon Emissions

Amid rising global concerns about climate change, significant international actions are being taken to promote efforts towards achieving net-zero greenhouse gas emissions by 2050. China has set a goal for carbon neutrality by 2060, the USA has recommitted itself to the Paris Agreement, and over 60 countries have joined the EU's efforts to reduce global warming by 55% by 2050 [154]. However, accurately tracking and reporting detailed carbon footprints from major greenhouse

gas sources remains a technical challenge, especially in complex supply chains that incorporate numerous independent processes such as production, harvesting, packaging, shipment, and retail with little to no vertical integration or information sharing among participants. The use of a central database to upload and extract carbon emission entries from data generated by different organizational domains through various processes is not feasible due to significant privacy and security concerns, as well as the burden of database maintenance [155].

Quantifying carbon footprints has become increasingly important due to its critical role in global warming. Carbon footprints, part of the broader ‘footprint family’ that includes ecological, energy, and water footprints, encompass direct and indirect CO_2 equivalent (CO_2eq) emissions from any system, process, or activity over a product’s lifecycle. For well-defined systems, carbon footprints are calculated using lifecycle assessment methods, considering emissions from raw material use to final disposal. Carbon footprint is quantified in CO_2eq units over a 100-year Global Warming Potential (GWP100) scale. For example, Methane (CH_4) has a GWP of 25, and Nitrous Oxide (N_2O) has a GWP of 265, meaning 25 parts of CH_4 or 265 parts of N_2O are equivalent to 1 part of CO_2 and their impact is accordingly higher. Carbon emissions are calculated as [156]:

$$E = A * EF(*GWP), \quad (6.1)$$

where E represents emissions in kg CO_2 , A represents activity that generates emissions in units of mass, volume or energy. EF represents the emission factor in kg CO_2eq per mass, volume or energy unit and GWP represents Global Warming Potential in kg CO_2eq .

Methane and other Greenhouse Gases (GHGs), such as nitrous oxide and fluorinated gases, have seen significant changes in their atmospheric concentrations over time, generally showing an increasing trend. For example, recent data indicates that methane’s (GWP) over a 20-year period is now estimated to be 84-87 times that of carbon dioxide, compared to older estimates of around 72 times [157, 158]. Likewise, the atmospheric concentration of nitrous oxide has increased, with its GWP now considered to be approximately 273 times that of carbon dioxide over a 100-year period [157]. A reference for major greenhouse gases and their potential impact on the environment is given in Table 6.1. Many academic works however, continue to use older established factors for

consistency with previous research. Despite the recent updates in the GWP of greenhouse gases, which are now understood to have a higher impact on climate change than previously estimated [157, 158], our measurements in the thesis continue to use the earlier well established factors. This approach ensures consistency and comparability with prior research, although it may not fully reflect the current scientific understanding of how these gases' affect climate change.

Greenhouse Gas (GHG)	Chemical Formula	Lifetime (years)	GWP (100-year)
Carbon Dioxide	CO ₂	Varies	1
Methane	CH ₄	12.4	25-36
Nitrous Oxide	N ₂ O	121	265-298
Hydrofluorocarbons (HFCs)	Varies	1.4-270	12-14,800
Perfluorocarbons (PFCs)	Varies	2,600-50,000	7,390-12,200
Sulfur Hexafluoride	SF ₆	3,200	23,500
Nitrogen Trifluoride	NF ₃	500	16,100

Table 6.1 Global Warming Potential (GWP) of various Greenhouse Gases (GHGs) over a 100-year period [159, 160]

The lifecycle of food products, particularly meat, greatly contributes to environmental degradation due to complex subsystems at each stage, such as pesticide use, refrigeration, and food disposal. The agricultural sector alone contributes 29% of all greenhouse gas emissions, with (CH₄) being a major component alongside (CO₂) and (N₂O). Livestock production, especially cattle raising, is a significant source of methane emissions during feeding and breeding, and land management and deforestation for grazing further add to emissions. Emissions from these activities are calculable at a fine-grained level but the lack of management platforms not controlled by any single organization is a major hurdle [161, 162]. Increasing global demand for animal protein has further led to more complex supply chains with numerous independent organizational participants.

The particular case of the beef supply chain which involves livestock management, feed harvesting, meat processing, cold storage, transportation, and retail, is important since all its stages are major greenhouse gas emitters. Hence, tracking and managing emissions from 'farm-to-fork' is challenging due to the independence of organizations along complex supply chains, as well as the lack of: (1) technology to identify, record and share data from potential emission sources, and (2) a decentralized and scalable regulatory management framework allowing organizations to connect

and collaborate [11].

6.2 Related Work on Emissions

Most studies on the beef supply chain's carbon footprint use lifecycle assessment but include only a subset of participants. They lack a comprehensive framework for detailed emission tracking [163, 164, 165]. Environmental impacts in supply chains have been studied using Lifecycle Assessment (LCA) methods, which quantify emissions and resource consumption relative to system output [166]. For beef supply chains, LCA can calculate carbon footprints and other impacts (e.g., energy use, global warming potential) at each stage, but disconnectivity between participants hampers tracking changes and their aggregated environmental effects. In addition to GHG protocols, standards like ISO 14040, 14044, 14046, 14064, and 14067 govern LCA methods, with bodies such as PAS 2050, IDF, IPCC, and FAO offer technical guidelines for quantifying carbon emissions [11, 156]. Though LCA is effective for evaluating environmental impacts from resource utilization, it is still subject to variability based on differing assumptions. LCA is therefore not universally applicable across systems unless there are less assumptions with more intuitive mechanisms to gather measurements from internal processes [167]. To cope up with the difficulty of detailed system measurements, guidelines for GHG emissions policy advocated by bodies like Intergovernmental Panel on Climate Change (IPCC) are commonly used. IPCC tier level 1 uses fixed emission factors for basic calculations of emissions, while tier levels 2 and 3 employ more detailed, country-specific, and regional data, respectively, to account for factors like fuel quality and technological differences [168]. Tiers 1 and 2 can also include level (or trend) and uncertainty assessments to identify significant emission variations over time. In our emissions framework, we use LCA parameters reported from tier 2 and 3 measures.

Today's food supply chains produce 13.7 billion metric tons of CO_2eq , about 26% of anthropogenic emissions, contribute to terrestrial acidification (32%), eutrophication (78%), and occupy 43% of arable land, using 87% for food and causing 90% of global water scarcity. Unaccounted large-scale cattle raising in the beef supply chain leads to significant deforestation, land degradation, and water loss, contributing 61% of food-related greenhouse gas emissions and 18% of total green-

house gases, with disconnected stakeholders making accountability difficult [44]. The modern beef supply chain includes complex subsystems from livestock management and feed harvesting to meat processing, cold storage, transportation, and retail, starting with calf rearing, followed by grain-fed breeding, and ending with beef distribution to retail stores and consumers [44]. In our BeefMesh framework, we consider a beef supply chain network which includes farmer, breeder, processor, distributor, retailer, and consumer, with a regulator overseeing tasks, allowing for variable distances and additional intermediaries to capture both local and extensive scenarios. The environmental impact of the beef supply chain is evaluated using a end-to-end method, includes breeding, feeding, processing, packaging, transportation, retailing, and cooking, with a focus on the carbon footprint of 1 pound of various beef cuts reaching the end consumer. Calculation of a beef supply chain's lifecycle inventory for carbon emissions is done by defining standard variables that represent every process (or event) at each participating supply chain organization.

6.3 The Carbon Tracking Application

To counter and manage carbon emissions related issues in beef supply chain, we utilize our proposed decentralized collaboration framework using blockchain and distributed databases that is configured to include beef supply chain specific organizations to allow tracking carbon emissions locally and globally (as shown in Figure 6.1). The flexibility to expand or shrink decentralized groups without disruption allows for automating a comprehensive and secure tracking of data originating from carbon-emitting sources throughout the chain. Controlled carbon information is subsequently harnessed by a federated entity (e.g., a regulator) that dynamically integrates and updates carbon conversion parameters agreed upon by all participants. Prior work on end-to-end carbon emission calculations for disjoint supply chains either relied on central databases for integrating required data from disparate participants with numerous assumptions, or focused on a restricted portion of the supply chain for their analyses. Our proposed collaboration framework enables mutual tracking, management, and regulation of emissions in a secure manner. It facilitates the formation of local or global emission group zones. Further benefits include the ability to develop and share sequestration solutions, as well as the federation and validation of green projects.

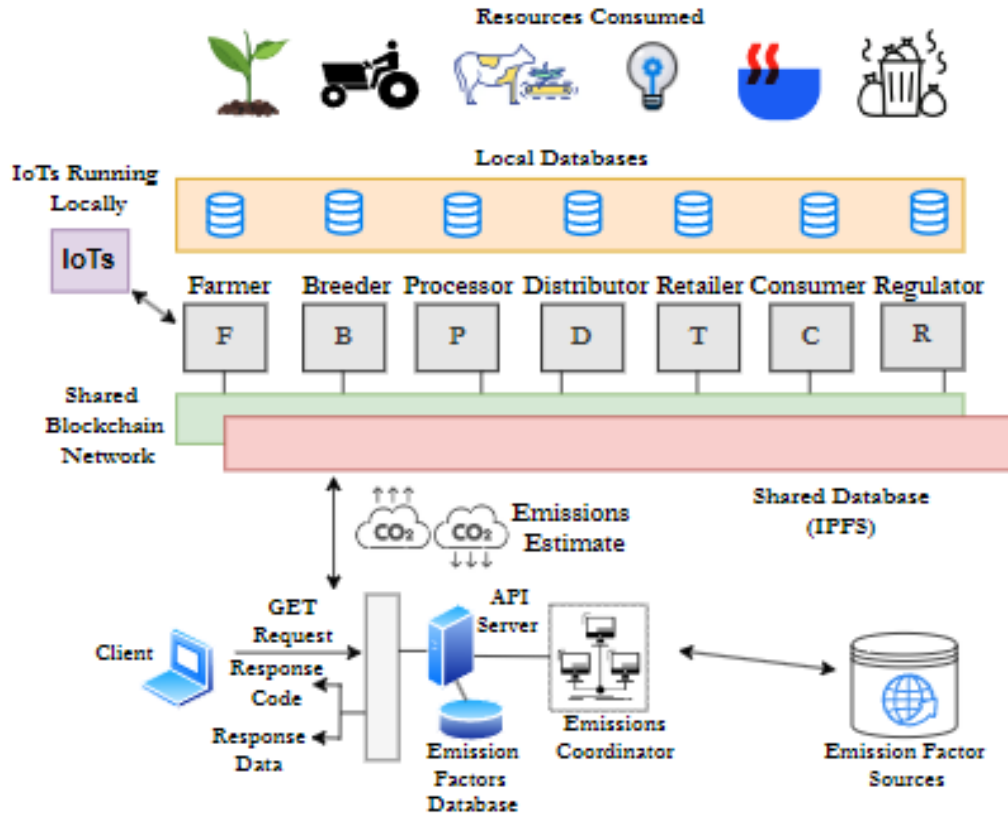


Figure 6.1 Major components of the emissions collaboration framework connecting fragmented supply chains participants

The carbon footprint tracking application is implemented by setting up local IoT and sensor container services to record and track resource consumption in different categories in each organization. We keep in view realistic scenario of 10 animals growing up at breeder for 15 months and moving through chain (as shown in Figure 6.2) when reporting proportion of resource that could be consumed. Organizations mutually setup private IPFS database nodes to store traceability records of emissions calculated from each organization when animals leave. Emissions are calculated from factors maintained and pulled by emissions server running as an independent organization with Create, Read, Update, and Delete (CRUD) operations and RESTful exposed services to which all group members connect. Any internal group member can also opt to serve as an emission factors manager. Emission factors are pulled from literature, NGO or government reporting sites and vetted by voting before being finalized to be used. This flexibility enables creation of local or global emissions zones with their own specific emission ranges. At the end, consumers can also

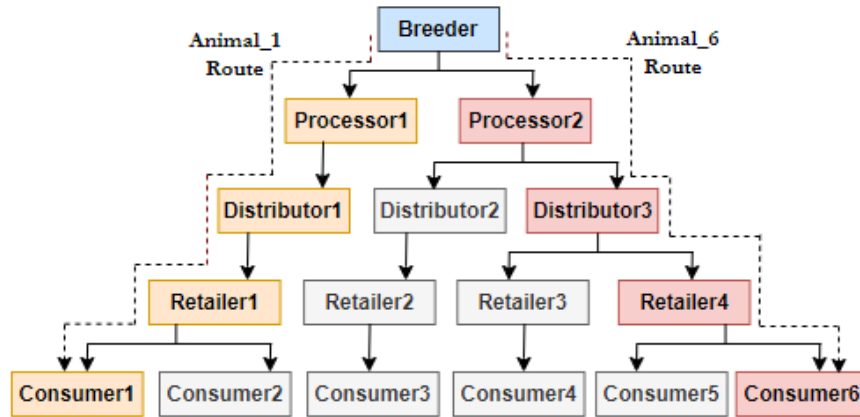


Figure 6.2 Routes taken by cattle in the beef supply chain. Starting from breeder, tracked animal (animal_1) takes the route highlighted on the left side and tracked animal (animal_6) takes the route highlighted on the right side

record distance travelled to buy beef package and method of cooking to get final last mile emissions along with ability to see per lb or per animal emissions. Hence, a multi-function peer-to-peer collaboration group is setup with organizations mutually controlling their shared data as shown in Figure 3.1. More details on starting a collaboration group and the services that run in each participating organization are described earlier in Chapter 3 and particularly in Section 3.5.

6.4 Internet of Things as the Enabler for Emissions Tracking

Organizations in a group download and spin IoT and sensor related containers locally that allow consuming data from various processes occurring in their domain. The containers use open source Mainflux software image to spin up numerous sensors and channel interfaces to consume, store and share data using IP standard. Considering the beef supply chain scenario, we focus on the sensors (energy, feed, by-products, packaging, plantation, fertilizers, pesticides, processes, cleaners and machinery) as shown in Table 6.2 (up till Table 6.6). This table summarizes factors used in our example to calculate final emission values for amount of resources consumed by organizations. The factors are maintained and pulled from an emissions server by mutual agreement (voting). An NGO, a regulator or any one of the participating organization can serve as emission factors maintainer. For each sensor, a number of channels are turned on to allow consuming data for different categories, e.g., for byproducts sensor, the channels could include but not limited to

(methane, manure, waste, blood). Finally, each group coordinates through collaboration channels to communicate and vote for the emissions calculation factor they will use for each category. This allows making groups that can cater to geographical local groups for emissions management, e.g., a local Michigan group using emissions factors extracted from LCA life cycle specific to Michigan region. Emission factors are coordinated and maintained using a flask-based RESTful service container supporting CRUD operations that either runs on a voted legitimate group member location or a coordinated new regulator organization domain similar to a collaborator/coordinator. To allow sensors to consume all types of data traffic, a number of messaging formats including HTTP, MQTT, CoAP, OPC-UA, and LoRa are configured and connected to the type of database most suitable for storing the particular data type (as shown in Figure 4.3).

Category	Emission Source	Unit Used	Comments/Description	CO ₂ Emissions Mentioned/Derived from Literature and Online Source
Energy	Electricity	kWh	Distributed over lines from national grid station	4.33×10 ⁻⁴ metric tons CO ₂ /kWh
	Diesel	lb	Assumes 100% of carbon is converted	10.180×10 ⁻³ metric ton CO ₂ /gallon
	Fossil	lb	From power plant using 95% coal	9.04×10 ⁻⁴ metric tons CO ₂ /pound
	Gasoline	lb	From kg CO ₂ per heat content of fuel	8.887×10 ⁻³ metric tons CO ₂ /gallon
	Natural Gas	feet ³	From natural gas burned as fuel and no gas related methane leakage	0.0053 metric tons CO ₂ /therm
	Steam	lb	From 80% efficient boiler using natural gas with 28btu input water (60F) and boiler-ratio of 1194mmBtu/lb steam	8.119×10 ⁻⁶ metric tons CO ₂ /gallon
Feed	Solar	kWh	Long term use with 15% efficiency	Offsets 50 grams of CO ₂ /kWh
	Wind Turbine	kWh	Average based on global coal-based	Offsets 6 grams of CO ₂ /kWh
	Alfalfa Hay	lb	Calculated from 89% dry matter	1 kg corresponds to 0.07 kg CO _{2eq}
	Distiller's Grain	lb	Calculated for dry matter maize/grain	1 kg corresponds to 859 g CO _{2eq}
	Corn/Maize	lb	Calculated for 32% dry matter of corn silage	1 kg corresponds to 0.14 kg CO _{2eq}
	Milk Replacer	lb	Amino acid balanced with whey protein	1 kg corresponds to 620 g CO _{2eq}
	Soybean	lb	Calculated from 87% dry matter	1 kg corresponds to 0.32 kg CO _{2eq}
	Vitamin/Mineral Mix	lb	From essential macro minerals and salt	1 kg corresponds to 500 g CO _{2eq}
	Protein/Fat Mix	lb	From microbial crude protein	1 kg corresponds to 750 g CO _{2eq}
	Grass Hay	lb	Calculated for 88% dry matter	1 kg corresponds to 0.15 kg CO _{2eq}
	Byproduct Waste	lb	Average of mixed dry matter	1 kg corresponds to 500 g CO _{2eq}
	Seeds	lb	Chia seeds used for omega-3 fatty acid	1 kg corresponds to 1.2 kg CO _{2eq}
	Barley	lb	Used in cereal beef systems for appetite	1 kg corresponds to 570 g CO _{2eq}
	Oats	lb	For increasing fiber and high hull in diet	1 kg corresponds to 570 g CO _{2eq}
	Wheat	lb	From 50% grain, 30% straw, 10% chaff	1 kg corresponds to 590 g CO _{2eq}
	Rye	lb	For early spring forage/ extended grazing	1 kg corresponds to 870 g CO _{2eq}
	Others	lb	Mix of Sorghum, rice, bran, millet, forage	1 kg corresponds to 500 g CO _{2eq}
Byproducts	Methane	lb	From belching due to enteric fermentation	220 pounds methane per cow/year 5500 pounds of CO _{2eq} per cow/year
	Manure	lb	From aerobic and anaerobic digestion	30000 g CO _{2eq} per tonne of storage
	Waste Discharge	lb	From volatile substance in feed mix, spoiled meat or manure dump	1 kg corresponds to 500 g CO _{2eq} 1.82 metric ton CO _{2eq} per gallon
	Blood Disposal	gal	Blood from abattoir has 18% volatile substance	216 mL methane per g of volatile substance, 1.82 metric ton CO _{2eq} /gallon

Table 6.2 Sources from regulatory platforms and literature used in calculating CO_{2eq} emissions

Category	Emission Source	Unit Used	Comments/Description	CO ₂ Emissions Mentioned/Derived from Literature and Online Source
Packaging	Plastic	kg	For cradle-to-grave plastic life cycle	1.7 kg CO ₂ per kg of plastic
	Paper	kg	Emissions from paper product creation	942 kg CO _{2eq} per metric ton paper
	Cardboard	kg	For cradle-to-grave cardboard life cycle	0.94 kg CO _{2eq} per kg of material
Plantation	Trees	ha	Assuming average sized 1000 trees/hactre	0.060 metric tons CO _{2eq} /urban tree
	Seeding	lb	Emissions from tilling and sowing seeds	1.17 kg CO _{2eq} /kg seeds sowed
	Liming	lb	From carbonates in dissolved lime	0.59 kg CO ₂ per kg lime application
Fertilizers	Nitrogen	lb	From potent nitrous oxide emissions	2.52 kg CO ₂ /kg of ammonium nitrate
	Potash	lb	From potassium chloride with 60% potash	0.23 kg CO ₂ /kg potash muriate
	Phosphate	lb	Emissions from di ammonium phosphate	0.73 kg CO ₂ /kg of phosphate
	Others	lb	Emissions From sulphur, gypsum, humus	0.5 kg CO ₂ /kg of product application
Pesticides	Fungicide	lb	From mixture of imidazole and trizole	3.9 kg CO ₂ /kg of mixed fungicide
	Herbicide	lb	Mixture of carbamate and biscarbamate	3 kg of CO ₂ /kg of mixed herbicide
	Insecticide	lb	Mixture of organophosphate	3.7 kg of CO ₂ /kg of mixed insecticide
Processes	Heating	kWh	Including heat transfer processes	0.19 kg CO _{2eq} /kWh of HVAC process
	Cooling	kWh	Includes emissions from refrigerants	0.19 kg CO _{2eq} /kWh of HVAC process
	Electro-chemical	kWh	From oxidation-reduction processes	0.25 kg CO _{2eq} per kWh
	Others	kWh	From chemical reactions that desorb CO ₂	0.19 kg CO _{2eq} /kWh of process
Cleaners	Cattle-Cleaner	lb	From cleaner containing glucamide	0.46 kg CO _{2eq} per kg of product
	Facility-Cleaner	lb	From cleaner containing cetyl-app Emissions from production and chemical reaction of mixture containing vinegar, borax, castile and disinfectants	5.16 kg CO _{2eq} per kg of product 0.7 lb CO _{2eq} / per lb cleaning agent
	Groundwater	Gal	From energy needed to deliver naturally replenished water source	0.22 g CO ₂ per L of ground water
	Brackish Groundwater	Gal	From energy needed for distribution	0.35 g CO ₂ per L of brackish water
	Desalinated Groundwater	Gal	From energy needed for desalination and distribution process	1.52 g CO ₂ per L of desalinated water
	Recycled Water	Gal	From recycling and distribution process	0.12 g CO ₂ per L of recycled water

Table 6.3 [Continuation of Table 6.2] Sources from regulatory platforms and literature used in calculating CO_{2eq} emissions

Category	Emission Source	Unit Used	Comments/Description	CO ₂ Emissions Mentioned/Derived from Literature and Online Source
Machinery	Pumps	kWh	Energy used for harvesting and cleaning	4.33×10 ⁻⁴ metric tons CO ₂ /kWh
	Fans	kWh	Energy used for regulating airflow	4.33×10 ⁻⁴ metric tons CO ₂ /kWh
	Site Transport	lb	Fuel burned for running vehicles	10.180×10 ⁻³ metric ton CO ₂ /gallon
	Materials Processing	kWh	Energy used for cutting/milling	4.33×10 ⁻⁴ metric tons CO ₂ /kWh
	Materials Handling	kWh	From equipment used for moving	4.33×10 ⁻⁴ metric tons CO ₂ /kWh
	Compressed Air	kWh	Used for pressurized cleaning	4.33×10 ⁻⁴ metric tons CO ₂ /kWh
	Electronics	kWh	From computers, sensors and readers	4.33×10 ⁻⁴ metric tons CO ₂ /kWh
	Others	kWh	Other equipment using renewable energy	Offsets 50 grams of CO ₂ /kWh
Consumption	Roast/Bake	lb	Based on using heated gas oven	6.97 kg CO _{2e} per kg of product
	Toast/Broil/Grill	lb	Based on using heated gas oven	4.91 kg CO _{2e} per kg of product
	Slow Cooker	lb	Based on using heated electric cooker	0.77 kg CO _{2e} per kg of product
	Deep Fry	lb	Based on using stove and frying oil	3.25 kg CO _{2e} per kg of product
	Steam	lb	Based on using stove for generating steam	3.28 kg CO _{2e} per kg of product
	Boil	lb	Based on using stove for boiling water	4.23 kg CO _{2e} per kg of product

Table 6.4 [Continuation of Table 6.2 and Table 6.3] Sources from regulatory platforms and literature used in calculating CO_{2eq} emissions

Category	Emission Source	Unit Used	Reference	Impact Factor
Energy	Electricity	kWh	[169]	Moderate
	Diesel	lb	[169]	High
	Fossil	lb	[169]	Very High
	Gasoline	lb	[169]	High
	Natural Gas	<i>feet</i> ³	[169]	Moderate
	Steam	lb	[170]	Moderate
	Solar	kWh	[171]	Negative
	Wind Turbine	kWh	[172]	Negative
Feed	Alfalfa Hay	lb	[173]	Low
	Distiller's Grain	lb	[174]	Moderate
	Corn/Maize	lb	[173]	Low
	Milk Replacer	lb	[175]	Low
	Soybean	lb	[173]	Low
	Vitamin/Mineral Mix	lb	[173]	Low
	Protein/Fat Mix	lb	[175]	Moderate
	Grass Hay	lb	[173]	Low
	Byproduct Waste	lb	[175]	Low
	Seeds	lb	[176]	Moderate
	Barley	lb	[177]	Low
	Oats	lb	[177]	Low
	Wheat	lb	[177]	Low
	Rye	lb	[177]	Moderate
	Others	lb	[175]	Low
Byproducts	Methane	lb	[178]	High
	Manure	lb	[179]	High
	Waste Discharge	lb	[175]	Moderate
		lb	[180]	Very High
	Blood Disposal	gal	[180]	Very High
Packaging	Plastic	kg	[181]	High
	Paper	kg	[182]	High
	Cardboard	kg	[183]	High
Plantation	Trees	ha	[169]	Negative
	Seeding	lb	[184]	High
	Liming	lb	[185]	Moderate
Fertilizers	Nitrogen	lb	[186]	Very High
	Potash	lb	[186]	Low
	Phosphate	lb	[186]	Low
	Others	lb	[186]	Low

Table 6.5 Reference of sources from regulatory platforms and literature used in calculating CO_2eq emissions

Category	Emission Source	Unit Used	Reference	Impact Factor
Pesticides	Fungicide	lb	[187]	Very High
	Herbicide	lb	[187]	Very High
	Insecticide	lb	[187]	Very High
Processes	Heating	kWh	[188]	Moderate
	Cooling	kWh	[188]	Moderate
	Electro-chemical	kWh	[189]	Low
	Others	kWh	[188]	Low
Cleaners	Cattle-Cleaner	lb	[190]	High
		lb	[190]	Very High
	Facility-Cleaner	lb	[191]	Very High
	Groundwater	Gal	[192]	Low
	Brackish Groundwater	Gal	[192]	Low
	Desalinated Groundwater	Gal	[192]	Low
Machinery	Recycled Water	Gal	[192]	Low
	Pumps	kWh	[169]	Moderate
	Fans	kWh	[169]	Moderate
	Site Transport	lb	[169]	High
	Materials Processing	kWh	[169]	Moderate
	Materials Handling	kWh	[169]	Moderate
	Compressed Air	kWh	[169]	Moderate
	Electronics	kWh	[169]	Moderate
Consumption	Others	kWh	[171]	Negative
	Roast/Bake	lb	[193]	High
	Toast/Broil/Grill	lb	[193]	High
	Slow Cooker	lb	[193]	Low
	Deep Fry	lb	[193]	High
	Steam	lb	[193]	High
	Boil	lb	[193]	High

Table 6.6 [Continuation of Table 6.5] Reference of sources from regulatory platforms and literature used in calculating CO_{2eq} emissions

Emission Source	Unit	Breeder B1	Processor		Distributor		
			P1	P2	D1	D2	D3
Electricity	kWh	50000	500	1000	0	0	0
Diesel	lb	5000	50	80	0	0	0
Fossil	lb	4000	0	0	0	0	0
Gasoline	lb	4500	30	50	3500	6000	10000
Natural Gas	<i>feet</i> ³	100000	500	1000	0	0	0
Steam	lb	200000	1000	2000	0	0	0
Bio Gas	<i>feet</i> ³	0	0	5000	0	0	0
Solar	kWh	0	0	50	0	0	0
Alfalfa Hay	lb	30000	500	1000	0	0	0
Distiller's Grain	lb	20000	0	0	0	0	0
Corn/Maize	lb	30000	0	0	0	0	0
Milk Replacer	lb	20000	0	0	0	0	0
Soybean	lb	5000	0	0	0	0	0
Vitamin/Mineral Mix	lb	10000	0	0	0	0	0
Protein/Fat Mix	lb	20000	0	0	0	0	0
Grass Hay	lb	30000	500	1000	0	0	0
Byproduct Waste	lb	10000	0	0	0	0	0
Seeds	lb	10000	0	0	0	0	0
Barley	lb	30000	0	0	0	0	0
Oats	lb	20000	0	0	0	0	0
Wheat	lb	10000	0	0	0	0	0
Rye	lb	10000	0	0	0	0	0
Others	lb	5000	0	0	0	0	0
Methane	lb	3000	2000	5000	0	0	0
Manure	lb	200000	3000	10000	0	0	0
Waste Discharge	lb	0	20000	45000	0	0	0
Blood Disposal	gal	0	200	500	0	0	0
Plastic	kg	50	50	150	5	6	6
Paper	kg	0	30	80	5	5	7
Cardboard	kg	0	50	150	10	12	11
Trees	ha	50	0	0	0	0	0
Seeding	lb	500	0	0	0	0	0
Liming	lb	500	0	0	0	0	0
Nitrogen	lb	4000	0	0	0	0	0
Potash	lb	2000	0	0	0	0	0
Phosphate	lb	1000	0	0	0	0	0
Others	lb	500	0	0	0	0	0
Fungicide	lb	100	0	0	0	0	0
Herbicide	lb	90	0	0	0	0	0
Insecticide	lb	120	0	0	0	0	0

Table 6.7 Resources consumed in the beef supply chain against animals movement and resultant *CO_{2eq}* emissions

Emission Source	Unit	Breeder B1	Processor		Distributor		
			P1	P2	D1	D2	D3
Heating	kWh	30000	50	100	0	0	0
Cooling	kWh	40000	100	250	400	700	1200
Electro-chemical	kWh	10000	0	0	0	0	0
Others	kWh	10000	0	0	0	0	0
Cattle-Cleaner	lb	500000	3000	7000	0	0	0
Facility-Cleaner	lb	100000	2000	5000	0	0	0
Groundwater	Gal	500000	10000	0	0	0	0
Brackish Groundwater	Gal	0	0	20000	0	0	0
Desalinated	Gal	0	0	0	0	0	0
Recycled Water	Gal	0	12000	30000	0	0	0
Pumps	kWh	10000	10	30	0	0	0
Fans	kWh	5000	10	30	0	0	0
Site Transport	lb	5000	10	25	0	0	0
Materials Processing	kWh	20000	20	40	0	0	0
Materials Handling	kWh	15000	20	40	0	0	0
Compressed Air	kWh	15000	5	15	0	0	0
Electronics	kWh	3000	5	15	0	0	0
Others	kWh	10000	0	0	0	0	0
Roast/Bake	lb	0	0	0	0	0	0
Toast/Broil/Grill	lb	0	0	0	0	0	0
Slow Cooker	lb	0	0	0	0	0	0
Deep Fry	lb	0	0	0	0	0	0
Steam	lb	0	0	0	0	0	0
Boil	lb	0	0	0	0	0	0
Transport	lb	0	500	1500	3500	6000	10000
Distance	mile	0	150	300	1200	2100	2900
Total Emissions From Organization (CO_2eq)	metric tons	356.109	4389.55	9882.69	3.824	6.545	10.89
Total Emissions Per lb of Beef (CO_2eq)	metric tons	0.051	0.3275	0.4315	0.0007	0.001	0.0015
Accumulated Emissions Per lb of Beef (CO_2eq)	metric tons	0.051	0.3756	.4825	0.3763	0.4835	0.484
Total Distance Traveled from Origin	mile	0	150	300	1350	2400	3200
Total Days Passed From Origin	days	548	552	553	556	558	600

Table 6.8 [Continuation of Table 6.7] Resources consumed in the beef supply chain against animals movement and resultant CO_2eq emissions

Emission Source	Unit	Retailer				Consumer					
		R1	R2	R3	R4	C1	C2	C3	C4	C5	C6
Electricity	kWh	100	200	350	400	0.1	0.3	0.5	1	1.2	1.5
Diesel	lb	0	0	0	0	0	0	0	0	0	0
Fossil	lb	0	0	0	0	0	0	0	0	0	0
Gasoline	lb	0	0	0	0	0	0	0	0	0	0
Natural Gas	<i>feet</i> ³	0	0	0	0	0	0	0	0	0	0
Steam	lb	0	0	0	0	0	0	0	0	0	0
Bio Gas	<i>feet</i> ³	0	0	0	0	0	0	0	0	0	0
Solar	kWh	0	0	0	0	0	0	0	0	0	0
Alfalfa Hay	lb	0	0	0	0	0	0	0	0	0	0
Distiller's Grain	lb	0	0	0	0	0	0	0	0	0	0
Corn/Maize	lb	0	0	0	0	0	0	0	0	0	0
Milk Replacer	lb	0	0	0	0	0	0	0	0	0	0
Soybean	lb	0	0	0	0	0	0	0	0	0	0
Vitamin/Mineral Mix	lb	0	0	0	0	0	0	0	0	0	0
Protein/Fat Mix	lb	0	0	0	0	0	0	0	0	0	0
Grass Hay	lb	0	0	0	0	0	0	0	0	0	0
Byproduct Waste	lb	0	0	0	0	0	0	0	0	0	0
Seeds	lb	0	0	0	0	0	0	0	0	0	0
Barley	lb	0	0	0	0	0	0	0	0	0	0
Oats	lb	0	0	0	0	0	0	0	0	0	0
Wheat	lb	0	0	0	0	0	0	0	0	0	0
Rye	lb	0	0	0	0	0	0	0	0	0	0
Others	lb	0	0	0	0	0	0	0	0	0	0
Methane	lb	0	0	0	0	0	0	0	0	0	0
Manure	lb	0	0	0	0	0	0	0	0	0	0
Waste Discharge	lb	30	40	60	80	0	0	0	0	0	0
Blood Disposal	gal	0	0	0	0	0	0	0	0	0	0
Plastic	kg	30	35	40	45	0	0	0	0	0	0
Paper	kg	30	35	40	45	0	0	0	0	0	0
Cardboard	kg	20	25	30	35	0	0	0	0	0	0
Trees	ha	0	0	0	0	0	0	0	0	0	0
Seeding	lb	0	0	0	0	0	0	0	0	0	0
Liming	lb	0	0	0	0	0	0	0	0	0	0
Nitrogen	lb	0	0	0	0	0	0	0	0	0	0
Potash	lb	0	0	0	0	0	0	0	0	0	0
Phosphate	lb	0	0	0	0	0	0	0	0	0	0
Others	lb	0	0	0	0	0	0	0	0	0	0
Fungicide	lb	0	0	0	0	0	0	0	0	0	0
Herbicide	lb	0	0	0	0	0	0	0	0	0	0
Insecticide	lb	0	0	0	0	0	0	0	0	0	0
Heating	kWh	0	0	0	0	0	0	0	0	0	0
Cooling	kWh	100	200	350	400	0.1	0.3	0.5	1	1.2	1.5
Electro-chemical	kWh	0	0	0	0	0	0	0	0	0	0
Others	kWh	0	0	0	0	0	0	0	0	0	0

Table 6.9 [Continuation of Table 6.7 and Table 6.8] Resources consumed in the beef supply chain against animals movement and resultant CO_2eq emissions

Emission Source	Unit	Retailer				Consumer					
		R1	R2	R3	R4	C1	C2	C3	C4	C5	C6
Cattle-Cleaner	lb	0	0	0	0	0	0	0	0	0	0
Facility-Cleaner	lb	0	0	0	0	0	0	0	0	0	0
Groundwater	Gal	5	10	20	30	0	0	0	0	0	0
Brackish Groundwater	Gal	0	0	0	0	0	0	0	0	0	0
Desalinated	Gal	0	0	0	0	0	0	0	0	0	0
Recycled Water	Gal	0	0	0	0	0	0	0	0	0	0
Pumps	kWh	10	10	10	10	0	0	0	0	0	0
Fans	kWh	10	10	10	10	0	0	0	0	0	0
Site Transport	lb	10	10	10	10	0	0	0	0	0	0
Materials Processing	kWh	20	20	20	20	0	0	0	0	0	0
Materials Handling	kWh	20	20	20	20	0	0	0	0	0	0
Compressed Air	kWh	5	5	5	5	0	0	0	0	0	0
Electronics	kWh	0	0	0	0	0	0	0	0	0	0
Others	kWh	0	0	0	0	0	0	0	0	0	0
Roast/Bake	lb	0	0	0	0	2	0	0	0	0	0
Toast/Broil/Grill	lb	0	0	0	0	0	4	0	0	0	0
Slow Cooker	lb	0	0	0	0	0	0	6	0	0	0
Deep Fry	lb	0	0	0	0	0	0	0	8	0	0
Steam	lb	0	0	0	0	0	0	0	0	10	0
Boil	lb	0	0	0	0	0	0	0	0	0	10
Transport	lb	0	0	0	0	10	20	30	40	50	60
Distance	mile	0	0	0	0	10	20	30	40	50	60
Total Emissions From Organization (CO_2eq)	metric tons	10.47	15.39	24.12	28.54	0.170	0.30	0.34	0.54	0.688	0.084
Total Emissions Per lb of Meat (CO_2eq)	metric tons	0.0174	0.019	0.02	0.02	0.0085	0.0076	0.0057	0.0068	0.0068	0.0083
Total Accumulated Emissions Per lb of Meat (CO_2eq)	metric tons	0.3930	0.5015	0.3963	0.3963	0.4015	0.4006	0.5082	0.4031	0.4031	0.4046
Total Distance Traveled from Origin	mile	1350	2400	3200	3200	1360	1370	2430	3240	3250	3260
Total Days Passed From Origin	days	561	573	620	625	566	567	579	627	633	634

Table 6.10 [Continuation of Table 6.7, Table 6.8 and Table 6.9] Resources consumed in the beef supply chain against animals movement and resultant CO_2eq emissions

6.5 Results and Discussion

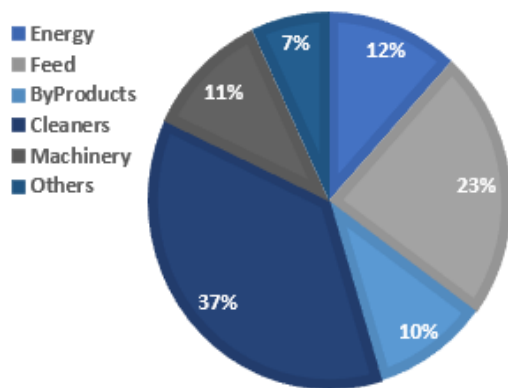
To test our proposed distributed collaboration framework for carbon emissions tracking, the BeefMesh application is extended to include a breeder, two processors, 3 distributors, 4 retailers, 6 consumers and 1 emissions server (emissions regulator) that coordinates maintaining emissions factors. Except for consumers, all organizations spin up their local IPFS, blockchain nodes, databases and IoT sensors. For consumer, only one instance of IPFS, and blockchain node is run at a dedicated location serving all consumers with a RESTful flask application to record their feedback or retrieve cattle public emissions traceability data available through a QR code. The whole setup is run over multiple IP reachable Virtual Machines (VMs) running Linux system (Ubuntu 22.04) with a minimum RAM of 8GB and 40GB Hard Disk. The setup can be run on a cloud as well as locally but each organization controls their own local setup of containers comprising blockchain nodes, distributed database node (IPFS), local IoT containers exposing sensors and channels along with a number of local databases to store resource consumption data.

Carbon emissions are calculated by tracking movement of 10 animals from end-to-end using data from 11 beef supply chain emissions categories (as shown in Table 6.1 till Table 6.6). For more details on emissions generated against different daily life activities, see reference [194]). Table 6.7 till Table 6.10 shows the amount of resources consumed with different units (under Unit column) as animals move from 1 Breeder (B1) to 2 Processors(P1, P2) and reach 6 consumers (C1-C6) through 3 Distributors (D1-D3) and 4 Retailers (R1-R4). Last 5 rows of Table 6.8 and Table 6.10 show total emissions piling up from left to right along with the days that have gone by as animals move through chain. We also track in detail two specific animals moving on different routes as shown in Figure 6.2. First through the collaborator, the required infrastructure of organizations is established along with necessary blockchain channels and privately connected IPFS network. Sensors and channels are set up for only the type of traffic that is expected in each organization, for example retailer organizations only need to capture electricity, wasted meat, packaging material used, refrigeration and other processes such as machinery used for cutting meat. For carbon emissions calculation, final aggregated data values are used locally or sent to a federated regulatory authority via secure

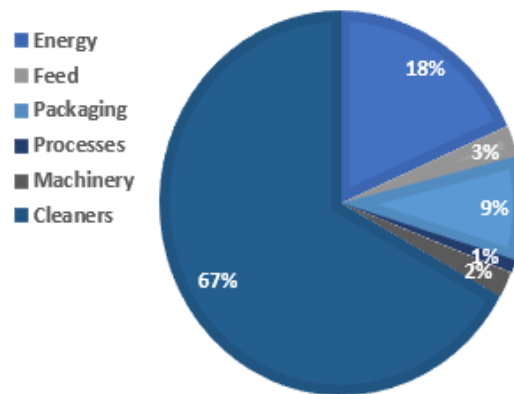
blockchain. An example is the final value of total feed consumed for 15 months at breeder. The framework makes it possible to calculate emissions at any instant (e.g., 1 minute of electricity use) by taking sensor records and retrieving total emissions against it from emissions server. The regulatory authority maintains a federated record of emission factors from vetted online resources (e.g., research articles). Vetting is done over blockchain ‘emissions-channel’ that can easily include an NGO overlooking local environment emissions. CRUD supported RESTful emissions server allows flexibility to experiment with the underlying factors for each emissions category, e.g., changing boiler efficiency rate to get a new heating emissions factor. Use of blockchain channels allows a secure and reliable way to maintain emissions for cross checking by regulators as animals move through chain.

We use a synthesized example of 10 animals on a farm to illustrate carbon emissions over time, tracking their physical characteristics, resource consumption, and carbon emissions, particularly for two animals (animal_1 and animal_6) using sensors. The final aggregated values over 18 months at breeder, shown in Table 6.8, highlight carbon emissions per lb of meat at 0.051 metric tons of CO_2eq . Key characteristics like weight, color, and age are also documented. For the ten animals, the weight in kilograms is {660, 663, 666, 669, 772, 775, 778, 882, 885, 888} and age in days {450, 480, 510, 540, 570, 600, 630, 660, 690, 720} is recorded at the end of 548 days leaving breeder. We consider a breeding ranch with a total are of 100 hectare with a planted area of 50 hectare.

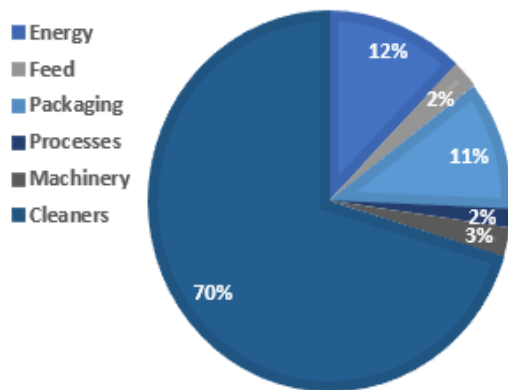
In our example, half the animals go to a smaller processing unit handling 40 animals daily (20 small, 20 large), processing 13,400 lbs of meat, and packaging 9,380 lbs. The other half go to a larger unit processing 100 animals daily, yielding 22,900 lbs of meat and packaging 16,030 lbs. Over three days, the carbon emissions per pound of meat are 0.3275 metric tons of CO_2eq for processor P1 and 0.4315 metric tons of CO_2eq for processor P2. Meat packages from two processing plants are distributed by three distributors, each traveling different distances and resulting in variable carbon emissions from fuel and cold storage. Detailed resource consumption is shown in Table 6.7 (and continued till Table 6.10), with distributor D1 delivering to retailer R1, distributor D2 to retailer R2, and distributor D3 to retailers R3 and R4. The final emissions per pound of meat are



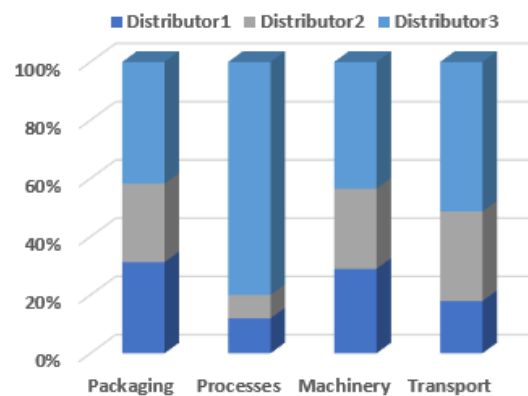
(a) breeder organization



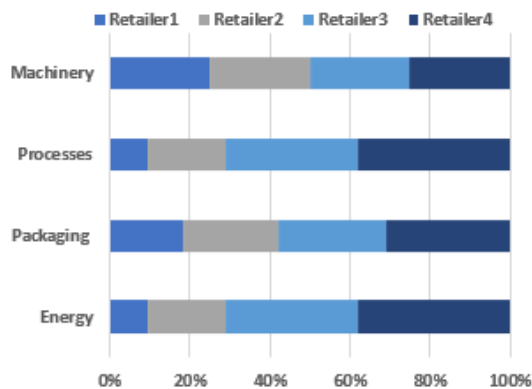
(b) processor_1 organization



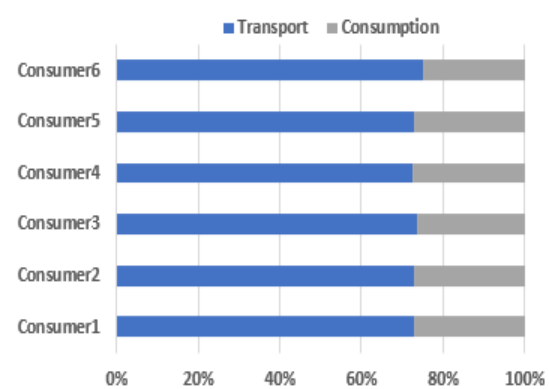
(c) processor_2 organization



(d) distributor organizations

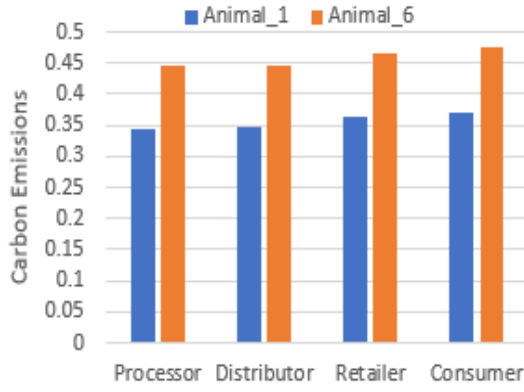


(e) retailer organizations

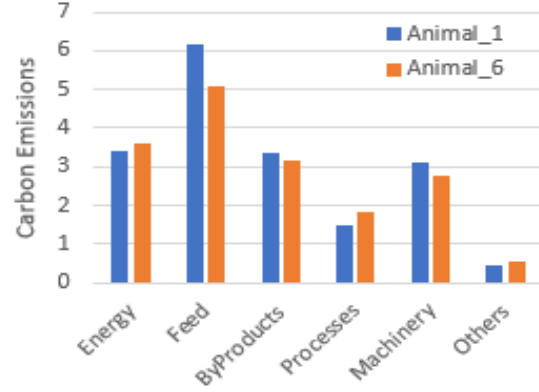


(f) consumer contribution

Figure 6.3 Proportion of metric tons of CO_2eq emissions contribution at different stages of beef chain



(a) at organization level

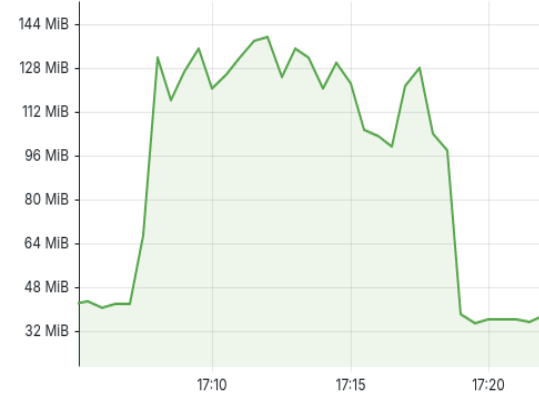


(b) at category level

Figure 6.4 Emissions contribution in metric tons of CO_2eq (y-axis) for animal_1 and animal_6 at different stages



(a) CPU utilization



(b) memory usage

Figure 6.5 Resource consumption for IoT application (a) y-axis shows percent of cpu use with timeline on x-axis (b) y-axis shows memory use in MiB with timeline (EDT) on x-axis

0.0007 metric tons of CO_2eq for D1, 0.001 metric tons for D2, and 0.0015 metric tons for D3.

The final processor P1 comes out to be 0.3275 metric tons of CO_2eq and 0.4315 metric tons of CO_2eq from processor P2. Retailers, the final step in the meat supply chain, use resources for processing, cold storage, and refrigeration, with meat often stored for days or months. For our example, emissions per pound of meat are 0.0174 metric tons CO_2eq for retailer R1 after 5 days, 0.019 metric tons for R2 after 6 days, 0.02 metric tons for R3 after 7 days, and 0.02 metric tons for R4 after 9 days, detailed in Table II. Consumer contribution to carbon emissions comes through travel and cooking methods. In our example, 6 consumers, each using a different cooking method and traveling varying distances to retail stores, contribute to final emissions of 401, 400, 508, 403,

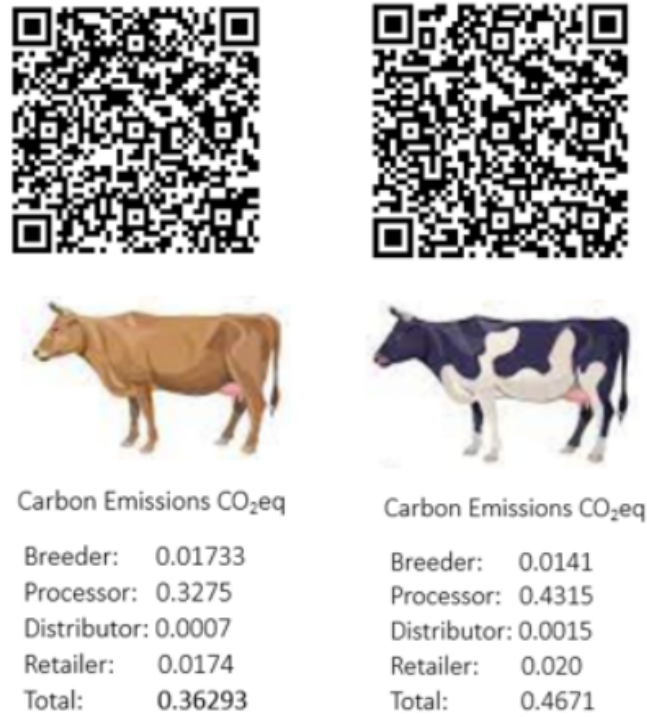


Figure 6.6 Emissions statistics pulled with QR code for (a) animal_1 on left side (b) animal_6 on right side

403, and 404 kg of CO_2eq per pound of meat, respectively, as detailed in Table 6.9 and Table 6.10.

A summary of proportion of carbon emissions generated throughout different stages of the beef supply chain as the 10 animals move from farm-to-fork is highlighted in Figure 6.3 (a-f). Figure 6.3(a), Figure 6.3(b) and Figure 6.3(c) shows percentage contribution of emissions from different categories for the 10 animals as the move across breeder and 2 processors (5 animals in each processor). Figure 6.3(d) and Figure 6.3(e) shows percentage of emissions taken by top contributing categories as same animal parts move through 4 distributors and end up being sold at 4 retailers. Figure 6.3(f) shows last mile emission contribution from consumers as they travel different distances to buy and cook beef. Figure 6.4 gives a consolidated view of individual animals contribution in carbon emissions for the whole supply chain. Figure 6.4(a) is a combined summary of the two precisely tracked animals contributing to emissions at organization level and Figure 6.4(b) is is their total contribution in metric tons of CO_2eq for different emission categories. The statistics in Figure 6.6 are embedded in QR code for consumers to decode (as shown in Figure 5.2).

The proposed carbon emissions tracking system allows for detailed tracking and management

of resource consumption across the chain. By tracking two animals from the same breeder through different routes involving processors, distributors, retailers, and consumers, detailed resource consumption can be reported. For example, emissions for animal_1 are 0.0173 per lb of CO_2eq at the breeder and 0.37143 per lb overall, while animal_6 has 0.0141 per lb at the breeder and 0.4754 per lb overall. At the end of its journey, accumulated emissions for animal_1 comes out to be 0.37143 per lb of CO_2eq and 0.4754 per lb of CO_2eq for animal_6.

To get an idea of system load from IoT application services, we continuously send sensor data over 10 channels for 10 minutes where each packet is roughly 1kb in size and store it in MongoDB database container. A minimum of 15 containerized services need to be run for IoT application which provides different functions such as authentication, databases, routing and message queuing. Even with 15 containers running for providing IoT services at its peak, combined CPU usage for the IoT containers averages around 2% (Figure 6.5(a)) and combined maximum memory usage averages around 130MiB (Figure 6.5(b)). The network transmission rate for the IoT containers averaged around 60kbps thereby providing lightweight functionality to accommodate other distributed containers and services.

6.6 Conclusion

Complex supply chains such as the beef chain significantly impact the environment through deforestation, water depletion, and carbon emissions. Tracking the carbon footprint is challenging due to the lack of vertical integration across stages like feed harvesting, processing, and retail. To address this, we utilized our proposed decentralized blockchain-based framework, BeefMesh, integrated with IoTs and databases to capture detailed emissions data throughout the supply chain, including transportation and waste management. This framework supports precise carbon emissions tracking and integrates diverse information sources while ensuring privacy and transparency. Using the distributed blockchain and IoT structure, we demonstrated its capability to securely enable data capturing, policy communication, facilitating reliable traceability and flexible environmental data sharing. Ultimately, this solution aims to promote emissions reduction and management across other complex food supply chains.

CHAPTER 7

OPTIMIZING RESOURCE UTILIZATION USING BEEFMESH FRAMEWORK

Note: The contents of this chapter, either in part or in full, are under submission to a conference. The authors of the manuscript in the order listed in the actual draft include Salman Ali, Cedric Gondro, Qiben Yan and Wolfgang Banzhaf.

The beef supply chain has a significant environmental impact, contributing to pollution and loss of bio-diversity. As it gets more complex with the involvement of numerous organizations, sharing of information between dispersed participants to track carbon emitting resource utilization becomes a challenge. The beef supply chain complexity and the lack of infrastructure for communicating data pertaining to carbon emissions hinders accurate measurement of resource utilization. To address the need for managing environmental degradation from excessive carbon emitting processes, we utilize our proposed decentralized collaboration framework, BeefMesh, and extend it to capture resource utilization that directly contributes to emissions. Using the emissions knowledge, we further optimize resource consumption across the beef supply chain using a collaborative approach.

7.1 Introduction

Due to the involvement of numerous dispersed and disjoint stakeholders, various environmentally toxic processes and activities in the beef supply chain keep poisoning the environment unnoticed. A first step towards the minimization of the toxic processes, particularly carbon emissions, is to connect scattered and disjoint parts of the supply chain such as production, processing,

harvesting, packaging, distribution, and retail [195]. Using a centralized platform to gather real-time emission measures for minimising them globally is not possible because of how the underlying emission sources are laid out throughout the supply chain with an additional privacy layer to avoid internal organizational components being exposed to outside actors. For the particular case of ‘beef supply chain’, LCA is considered a valuable tool to manage carbon footprints, as it is instrumental in identifying emission hot spots, that can ultimately lead to devising strategies for its mitigation. A summary of LCA and related work on calculating carbon emissions in different systems, including supply chains, has been summarized in Section 6.2.

Considering the measurement of carbon footprint as the first step, significant steps are required to be able to put them into use for environmentally sustainable supply chain processes. Recent analyses has highlighted the importance of optimizing logistics and transportation to reduce carbon footprints, as long-distance ‘food miles’ significantly contribute to greenhouse gas emissions [196]. Sustainable agricultural practices, such as precision farming and renewable energy usage have also shown potential for emission reductions [197]. However, the literature shows that the fragmented nature of the food supply chain complicates carbon optimization, necessitating collaborative frameworks, transparent communication, and supportive policy interventions [198, 199]. Any method for optimizing emissions is ineffective until there is a framework on which dispersed participants can trust to securely let it gather internal organizational statistics for sharing and benefiting from. A more realistic approach for optimizing emissions in a supply chain is to use an application that provides clients, control over the information they are sharing, how they are sharing, where it is stored and how it is incorporated into policies .

To address the challenges in collecting carbon emissions related data and converting it into knowledge that can help to optimize resource consumption and minimize emissions, we utilize our proposed collaboration framework described in detail in Chapter 3 and Chapter 4. In particular, the carbon emissions are calculated using the BeefMesh application infrastructure described in Section 6.3. The infrastructure is further extended to enable a carbon emissions optimization application for the beef supply chain network. The system boundaries for our application scenario

is shown in Figure 7.1. The emissions optimization application is meant to highlight the difficulty and importance in bringing together participants in supply chains with high climate impact for the purpose of collaborating, sharing vital information and optimizing internal functions and decisions for the greater environmental good. With the ability to optimize emissions in a supply chain, it becomes easier to develop sequestration solutions and policies in addition to validating and adopting environment friendly green projects.

7.2 The Resource Consumption Optimization Application

An extended BeefMesh application for resource optimization is set up by coordinating through a group initiator server to demonstrate emission tracking and optimization application (as described earlier in Section 3.5). A collaboration group comprising of breeder, processor (abattoir), distributor, retailer and an emissions management and optimization organization. Due to resource restrictions (number of physical machines with unique IP addresses) and for demonstration purposes, we set up the organizations in such a way that they can be used as forking points to represent more than one organization for illustrating a setup of hundreds of participants. Specifically we set up 3 nodes of each organization (breeder, processor, distributor, retailer) and use multiple blockchain channels (e.g., breeder-channel-N) to represent different organizations (e.g., breeders) participating. When the optimization problem becomes more complex (e.g., requiring hundreds of breeders), we further reconfigure each blockchain channel to represent more than one organization by re-using underlying variables defined by the chaincode (program installed on blockchain). This stems from our limitation to arrange and manage hundreds of physical machines (or VMs) at one place at a time or to buy costly VM instances on cloud. In practice however, a simple lightweight VM (host machine) is good enough to run services intended to be run at each organization (blockchain node, IPFS, local databases and local IoT interfaces). Each machine (VM) used in experiments utilizes Linux (Ubuntu 22.04) with at least 6GB of RAM and 40GB of hard disk space. This configuration can be deployed both on cloud platforms and locally, with each organization managing its own local setup of containers, including blockchain nodes, distributed database nodes (IPFS), IoT containers with sensors and channels, and various local databases for storing resource consumption data.

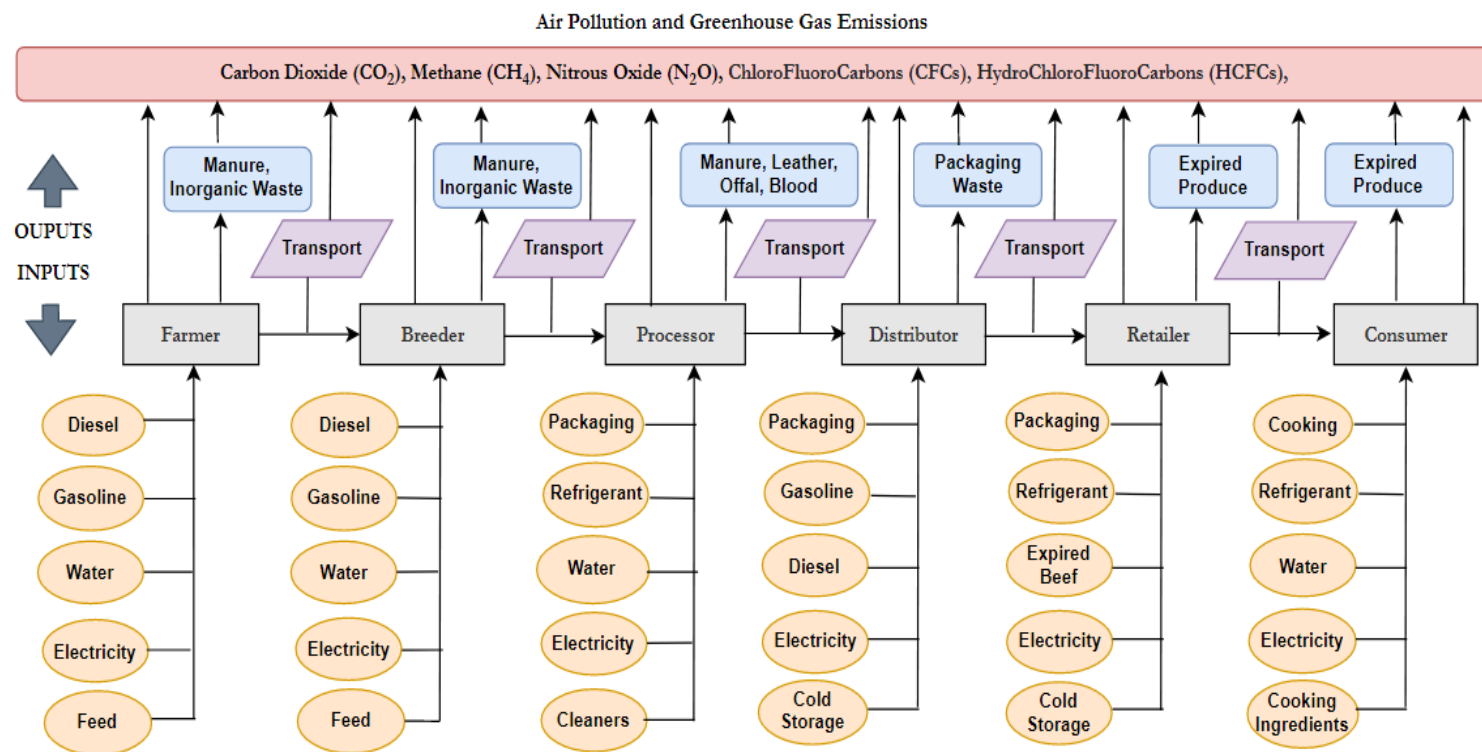


Figure 7.1 System boundaries of proposed carbon optimization application

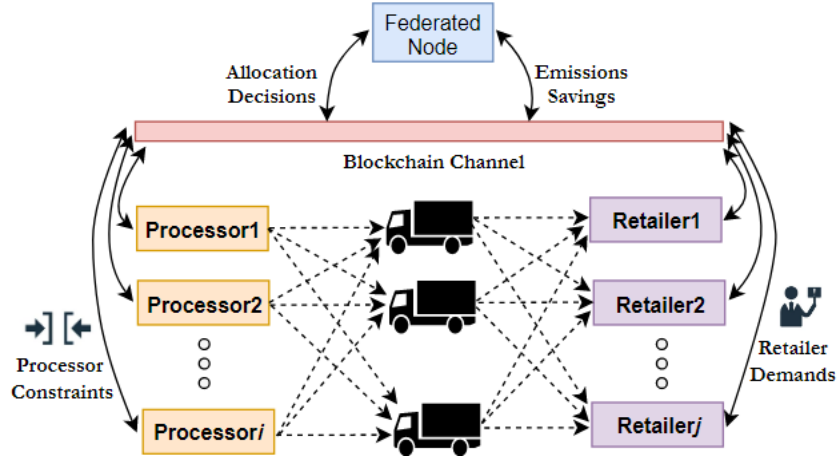


Figure 7.2 Decisions on optimal animal paths are shared over blockchain channels

Carbon emissions tracking, optimization and related decisions are enabled by gathering and utilizing resource consumption data from IoT and sensor applications running at each organization in a group (as described in detail in Section 6.3.1). The resource consumption data (e.g., total electricity usage at the end of period) used and reported in Table 6.7 (and continued till Table 6.10) are non-sensitive information that do not expose underlying details of organizations. Nevertheless organizations can decide to not share resource consumption at the end of period, rather sharing final emissions against each major category (e.g., *Energy* category). As animals move from one end to the other, emissions data gets recorded in private and public ledgers (blockchain channels) depending upon the collaborating groups agreement. For our specific beef chain application, public data allows regulators to have a broad overview of emissions generated at each organization per lb of beef for specific animal in metric tons CO_2eq (refer to example shown in Table 6.7 till Table 6.10). Use of distributed databases (IPFS) allows an immutable platform for storing timeline sequence of emission events at each stage of supply chain. Hence, we set up a distributed IPFS database storage with private settings and ideal for tractability to allow users to maintain records as processes and events unfold in the chain (as shown in Figure 3.6). The CID from records uploaded on IPFS are stored in blockchain providing a one-to-one mapping for organizations and regulators to verify data and events. Organizations also start other database containers locally (SQL and NoSQL) to create a data pipeline where raw data can be stored first before being sorted out to put on blockchain and IPFS (as shown in Figure 3.6). IoT databases further allow filtering and formatting

data at application level based on HTTP, MQTT, CoAP, LoRa and OPC-UA configurations to be compatible for storing (as shown in Figure 4.3).

Together with the combination of IoTs/sensor interfaces running at each organization, distributed IPFS and blockchain channels set up throughout the collaborating group, emissions from resource consumption are recorded and managed for processes such as energy consumption, feed production and waste management. With the distributed and decentralized nature of the framework, local information (e.g., internal details of animals and fine-grained details of resource consumption) is kept within organization and global information is shared using mutually managed databases and common blockchain channels. A lightweight GO language-based program is installed on all blockchain channels allowing to handle information consistently. The program (chaincode) allows all native operations like reading, writing, updating and deleting records with different formats (strings, characters,) based on the level of ownership in addition to collaborating on policies and decision making. With one run of emissions calculation from end-to-end, a group can form a reference framework for estimating possible future emissions when a different set of animals move through the same route. This generates the possibility of optimizing the supply chain for lesser emissions before hand and creating specific routes for animals depending on demand and supply. The framework is flexible enough to reconfigure a defined reference of emissions for a group when any of the organization makes major changes internally (e.g., use of solar panels and greener methods). Optimization calculations and decisions can be made by incorporating a mutually managed third party organization (scientific application node) in the group running the same micro services (IPFS, blockchain, local database) but running optimization algorithms on demand and supply matrices with constraints pulled from emissions reference framework for the group.

7.3 Results and Discussion

The carbon optimization (reduction) problem in our example is defined as a federated machine learning decision process (as shown in Figure 7.2). A set of distributed nodes (a group) representing source and destination organizations decide on which routes for animals to send that would minimize

emissions. The set of possible choices from which each organization can decide a route is sent to a federated node (called optimizer from here onwards). Optimizer node also maintains a reference framework (total possible emissions for each route) by making use of resource consumption for each organization in the path and its contribution to emissions (in metric tonnes of CO_2eq per lb of beef). Table 6.2 (and continued till Table 6.6) is used as reference to calculate emissions from resource consumption. The optimizer node forms a linear programming model from presented choices and runs linear programming models over it until an solution is found. Decisions are then sent to requesting nodes. Take the case of a number of processors trying to decide which retailers should be chosen. The carbon emissions cost matrix C_{ij} represents emissions cost incurred when beef is shipped from processor i to retailer j . Emissions cost takes into consideration the resources (e.g., refrigeration, fuel etc) consumed for the travel distance between processor and retailer. In essence, behind every emissions cost is a list of resources consumed (as shown in Figure 7.3). Emissions cost can be directly converted to financial costs, resulting in possible savings. Consider each retailer with a demand of beef quantity which can be expressed as D_{R_j} while each processor has a limit of beef production during the specified time (S_{P_i}). The decision variables can then be defined as X_{ij} where X_{11} represents amount of beef that can be delivered from processor 1 to retailer 1, X_{12} represents amount of beef that can be delivered from processor 1 to retailer 2 and so on. The objective function then takes the form:

$$Minimize(\sum_{i=1}^n \sum_{j=1}^m C_{ij} * X_{ij}) \quad (7.1)$$

subject to processor and retailer constraints

where main objective is to choose quantity of beef that can be supplied from a processor to a retailer while minimizing overall carbon emissions across all suppliers and retailers. Optimization problem is therefore the sum product of carbon emissions cost matrix and the allocation matrix. Each carbon cost entry (c_{ij}) in the Cost matrix is an aggregation of resultant carbon emissions from all resources consumed when a given amount of beef is processed and shipped. Carbon cost entry

in the Cost matrix can therefore be defined as:

$$\begin{aligned}
C_{ij} = & c_{energy} + c_{feed} + c_{byproducts} + c_{packaging} + c_{fertilizers} \\
& c_{pesticides} + c_{processes} + c_{cleaners} + c_{machinery} \\
& - c_{plantation} - c_{sequestration}
\end{aligned} \tag{7.2}$$

The constraints for objective function are defined in terms of total capacity of processors supply across all retailers and the retailers total demand across all processors. The processor related constraint can be defined as:

$$\begin{aligned}
X_{11} + X_{12} + X_{13} + X_{14} + \dots + X_{1j} & \leq T_{p_1} \\
X_{21} + X_{22} + X_{23} + X_{24} + \dots + X_{2j} & \leq T_{p_2} \\
X_{31} + X_{32} + X_{33} + X_{34} + \dots + X_{3j} & \leq T_{p_3} \\
& \vdots \\
X_{i1} + X_{i2} + X_{i3} + X_{i4} + \dots + X_{ij} & \leq T_{p_i}
\end{aligned} \tag{7.3}$$

Processor related constraints in essence highlight that the total allotment of beef by weight done across all retailers for a given processor or i – th abattoir cannot be more than the capacity of the processor (abattoir). Retailer related constraints can then be defined as:

$$\begin{aligned}
X_{11} + X_{21} + X_{31} + X_{41} + \dots + X_{i1} & \geq T_{R_1} \\
X_{12} + X_{22} + X_{32} + X_{42} + \dots + X_{i2} & \geq T_{R_2} \\
X_{13} + X_{23} + X_{33} + X_{43} + \dots + X_{i3} & \geq T_{R_3} \\
& \vdots \\
X_{1j} + X_{2j} + X_{3j} + X_{4j} + \dots + X_{ij} & \geq T_{R_j}
\end{aligned} \tag{7.4}$$

where the constraints define the total allotment of beef by weight to each retailer or the $j - th$ retailer variable should be set such that the retailers demand is met. To make scenario realistic, decision variables are set to take only positive integer values, hence making it an ‘Integer Linear Programming’ problem. The decision variable related allocation matrix is then defined as:

$$\begin{aligned}
 & [[X_{11} \ X_{12} \ X_{13} \ X_{14} \ ... \ X_{1j}] \\
 & [X_{21} \ X_{22} \ X_{23} \ X_{24} \ ... \ X_{2j}] \\
 & \vdots \\
 & [X_{i1} \ X_{i2} \ X_{i3} \ X_{i4} \ ... \ X_{ij}]] \quad (7.5)
 \end{aligned}$$

For demonstration of our framework’s usefulness, we present a number of linear optimization problems. The optimizer node gathers data and sends back decisions to requesting pair of nodes through blockchain channels. Defined problems involve minimizing carbon emission costs between (1) Breeder-Processor (2) Breeder-Distributor (3) Breeder-Retailer (4) Processor-Distributor (5) Processor-Retailer and (6) Distributor-Retailer. Each problem requires possible resource consumption estimates (reference) between multiple source and destination pairs before the actual carbon emission cost matrices can be formed and used for optimization. Since there is a linear trend of resource consumption and carbon emissions output, minimizing carbon emissions results in minimizing resource consumption with possible savings. Formulated optimization problems at the optimizer node is solved by utilizing open source PuLP library for Python. PuLP supports a number of solvers, including the CPLEX, CBC and GUROBI solvers, which we employed in our computations.

To begin with, a simplified example of 6 organizations (2 processors, 4 retailers) is presented first. Excluding carbon emissions at processors, the target is to finalize a joint decision for allocation of resources such that retailers demands are met within processors constraints. Without taking into account emissions from processors, the emissions for supplying beef from processors to retailers are a direct result of using packaging materials (plastic, cardboard an paper), cooling process and

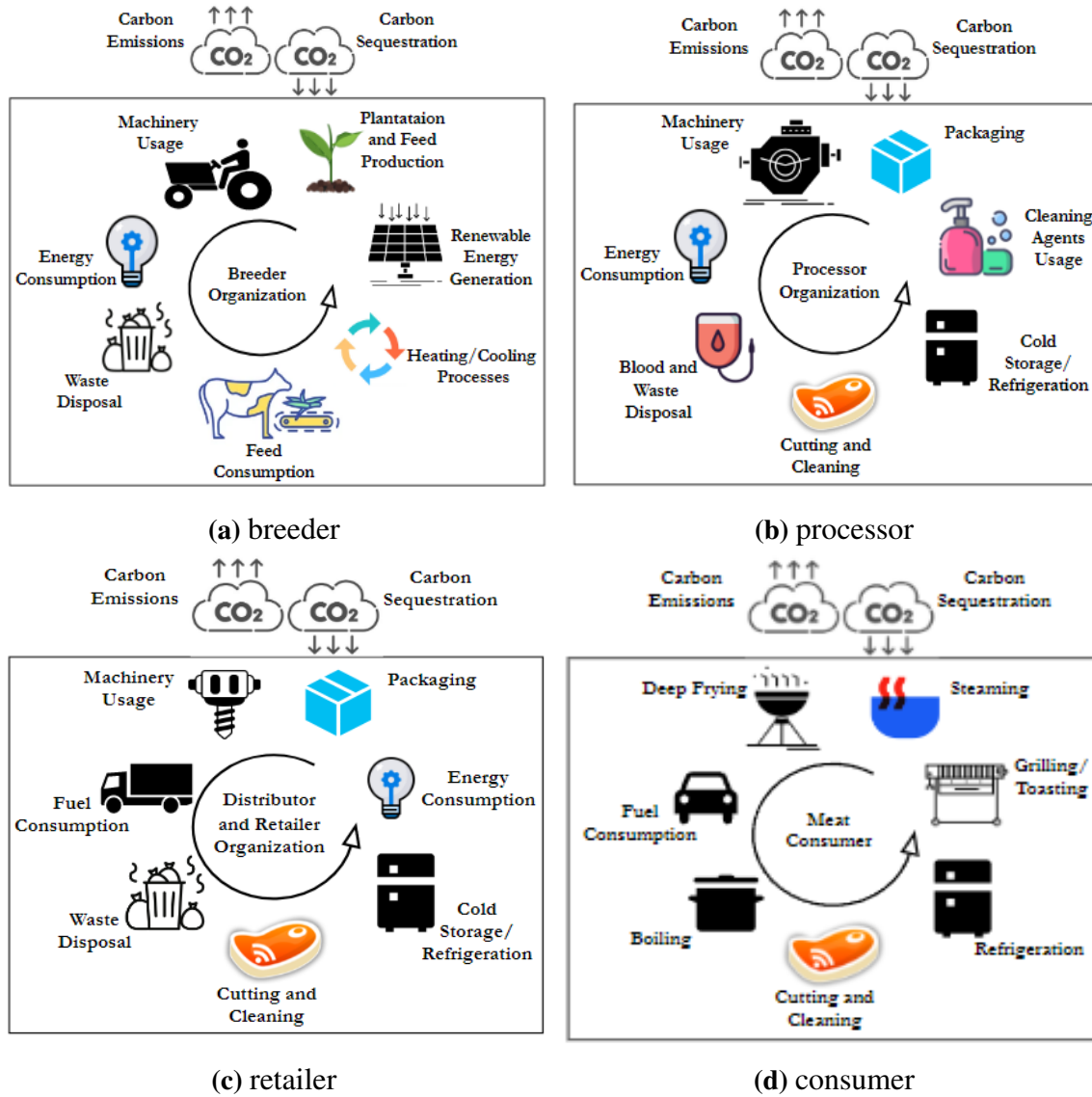


Figure 7.3 Major source of emissions at (a) breeder (b) processor (c) retailer, and (d) consumer side

use of fuel (gasoline, diesel) for transportation. Considering emission factors, major contributing in emissions here comes from fuel which is directly proportional to the distance between processor and retailer. In the example, processor constraints (P_i), retailer demands (R_j) and emissions cost matrix (C_{ij}) are:

$$P_i = \begin{bmatrix} p_1 & p_2 \end{bmatrix} = \begin{bmatrix} 16052.6 & 15986.4 \end{bmatrix} \quad (7.6)$$

$$\begin{aligned}
R_j &= \begin{bmatrix} r_1 & r_2 & r_3 & r_4 \end{bmatrix} \\
&= \begin{bmatrix} 6060.1 & 7456.6 & 5158.7 & 5042 \end{bmatrix}
\end{aligned} \tag{7.7}$$

$$\begin{aligned}
C_{ij} &= \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \end{bmatrix} \\
&= \begin{bmatrix} 12.93 & 4.87 & 8.38 & 6.93 \\ 10.54 & 11.75 & 14.02 & 10.87 \end{bmatrix}
\end{aligned} \tag{7.8}$$

where i represents processor and j represents retailer and $C_{ij} = c_{energy} + c_{packaging} + c_{processes}$. Each individual cost variable from the above equation can be expanded as an aggregation of emissions as following:

$$c_{11} = 12.632 + 0.0387 + 0.259 \approx 12.93 \text{ } CO_2eq$$

$$c_{12} = 4.76 + 0.015 + 0.097 \approx 4.87 \text{ } CO_2eq$$

$$c_{13} = 8.19 + 0.025 + 0.167 \approx 8.38 \text{ } CO_2eq$$

$$c_{14} = 6.77 + 0.02 + 0.14 \approx 6.93 \text{ } CO_2eq$$

$$c_{21} = 10.30 + 0.031 + 0.21 \approx 10.54 \text{ } CO_2eq$$

$$c_{22} = 11.48 + 0.035 + 0.23 \approx 11.75 \text{ } CO_2eq$$

$$c_{23} = 13.70 + 0.042 + 0.28 \approx 14.02 \text{ } CO_2eq$$

$$c_{24} = 10.62 + 0.033 + 0.22 \approx 10.87 \text{ } CO_2eq$$

$$\text{where, } C_{ij} = c_{energy} + c_{packaging} + c_{processes} \quad (7.9)$$

Source	Sink	Total (i) Sources	Total (j) Sinks	Supply Matrix (i x j)			Demand Matrix (j x i)		
				mean	median	st. dev.	mean	median	st. dev.
Breeder	Processor	30	50	23490.06	21292.75	10188.31	8055.68	7366.3	2276.99
Breeder	Processor	100	200	23811.50	23500.4	8280.4	9042.649	9141.9	2270.09
Breeder	Processor	300	500	24925.98	24422.94	8591.33	9000.02	9096.90	2279.79
Breeder	Distributor	30	50	35756.50	34682.60	8145.69	10786.43	10647.6	2339.71
Breeder	Distributor	100	200	34142.03	32998.75	8273.56	10955.40	11138.05	2262.13
Breeder	Distributor	300	500	35177.19	35033.39	8865.77	10995.55	11005.6	2293.27
Breeder	Retailer	30	50	29260.83	29550.0	6454.88	12400.45	12299.4	1324.63
Breeder	Retailer	100	200	29523.92	29826.25	5686.11	12586.97	12608.55	1495.45
Breeder	Retailer	300	500	29939.36	29577.4	5701.24	12441.69	12356.85	1453.18
Processor	Distributor	30	50	35416.80	35522.1	2737.91	17783.64	17963.2	1383.86
Processor	Distributor	100	200	35222.81	35144.5	2649.52	17609.90	17740.55	1408.52
Processor	Distributor	300	500	34978.16	34891.2	2836.43	17478.80	17520.3	1366.39
Processor	Retailer	30	50	17757.84	17844.45	1626.50	7200.43	6927.55	1445.52
Processor	Retailer	100	200	17472.58	17226.1	1551.86	7601.64	7748.15	1446.26
Processor	Retailer	300	500	17408.02	17316.9	1445.16	7543.87	7613.15	1451.33
Distributor	Retailer	30	50	24566.70	24499.15	3014.47	12328.26	12108.40	1510.19
Distributor	Retailer	100	200	25465.63	25704.8	2817.76	12381.48	12369.95	1439.20
Distributor	Retailer	300	500	25204.84	25347.55	2875.90	12440.57	12503.95	1482.90

Table 7.1 Linear optimization problems are formulated between Breeder-Processor, Breeder-Distributor, Breeder-Retailer, Processor-Distributor, Processor-Retailer and Distributor-Retailer. Each problem consists of a supply matrix consisting of maximum amount of beef in pounds (lbs) that can be supplied from the destination and a demand matrix that represents the required amount of beef in pounds (lbs) at the source. The supply and demand matrix properties are reported for beef quantity in pounds (lbs). The carbon cost matrix properties are reported for carbon emissions in metric tonnes of CO_2eq . The objective value for optimization algorithm is reported in quantity of beef in pounds (lbs) [194]. The total decision variables are the sum of assigned variables and the ones that are not assigned

Source	Sink	Carbon Cost Matrix (i x j)			Objective Value	Decision Variables	
		mean	median	st. dev.		Assigned	Not Used
Breeder	Processor	5491.47	5473.22	1430.17	1280253910.04	59	1441
Breeder	Processor	5493.87	9042.649	1442.07	2813051956.04	124	4876
Breeder	Processor	5501.26	5504.47	1441.98	13587691271.41	608	149392
Breeder	Distributor	8589.07	8595.94	1967.09	2855380690.65	51	1449
Breeder	Distributor	8494.70	8479.43	2014.61	11109356795.25	232	19768
Breeder	Distributor	8506.71	8505.25	2020.31	27628542918.33	564	149436
Breeder	Retailer	6550.42	6566.02	1449.21	2603315818.27	65	1435
Breeder	Retailer	6513.15	6546.71	1440.60	10218366212.98	266	19734
Breeder	Retailer	6501.41	6495.53	1444.75	24990253277.20	616	149384
Processor	Distributor	4997.86	4964.85	1148.90	2814876618.21	72	1428
Processor	Distributor	4998.53	4998.64	1159.60	10751186876.66	299	19701
Processor	Distributor	5003.39	5000.40	1154.90	26357738224.01	684	149316
Processor	Retailer	4454.42	4466.26	1438.02	782607354.74	60	1440
Processor	Retailer	4498.68	4488.10	1444.13	3122988567.11	271	19729
Processor	Retailer	4499.32	4494.32	1443.43	7614762168.17	633	149367
Distributor	Retailer	10.03	10.09	2.88	3290376.09	69	1431
Distributor	Retailer	10.02	10.08	2.89	12686037.67	296	19704
Distributor	Retailer	10.00	10.01	2.88	31345580.87	687	149313

Table 7.2 [Continuation of Table 7.1] The results from linear optimization problems formulated between Breeder-Processor, Breeder-Distributor, Breeder-Retailer, Processor-Distributor, Processor-Retailer and Distributor-Retailer organizations

Each individual carbon emissions cost variable can be converted to financial costs and vice versa for any organization. For example, carbon emissions cost for processor 1 and retailer 1, $c_{11} = 12.632 + 0.0387 + 0.259$ represent financial costs incurred on fuel, packaging and cooling as follows. Considering, a truck using 6000 lb of gasoline produces 6.37 CO_2eq , a distributor generating 12.632 CO_2eq from gasoline will use 11898 lb of gasoline which is \$3566 considering \$2.5 per gallon. With an average truck traveling 2100 miles on 6000 lb of gasoline (approximately 3 miles per gallon), total distance traveled would be 4161 miles which is the second longest distance in our example. Considering 700KWh produces 0.133 CO_2eq of emissions, 0.259 CO_2eq emissions would equate to 1363.11 KWh. With an average cost of 20 cents per KWh of energy use, 1363.11 KWh would equate to \$27.3. Considering 20-40-40% emissions from paper, cardboard and plastic, 0.0387 CO_2eq of packaging emissions can be broken down into 0.0155 CO_2eq from use of cardboard, 0.0155 CO_2eq from plastic and 0.0077 CO_2eq from paper. With 6kg of plastic producing 0.01 CO_2eq of emissions, 0.0155 CO_2eq of emissions equate to 9.3kg of plastic. With

per kg cost of \$0.5, 9.3kg of plastic would cost \$4.65. With 12kg of cardboard producing 0.0113 CO_2eq of emissions, 0.0155 CO_2eq of emissions equate to 16.46kg of cardboard. With per lb cost of \$0.1, 16.46kg of cardboard would cost \$3.63. With 5kg of paper producing 0.0047 CO_2eq of emissions, 0.0077 CO_2eq of emissions equate to 8.19kg of paper. With per kg cost of \$0.9, 8.19kg of paper would cost \$7.91. Hence, total financial cost associated with the carbon emissions cost for c_{11} would be: $f_{11} = 3566 + 4.65 + 3.63 + 7.91 + 27.3 = \3599.5 .

Going back to the example of 6 organizations, given carbon emissions cost matrix, the resource allocation matrix with decision variables is:

$$\begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \end{bmatrix} \quad (7.10)$$

The processor related (supply) constraint are defined as:

$$\begin{aligned} X_{11} + X_{12} + X_{13} + X_{14} &\leq 16052.6 \\ X_{21} + X_{22} + X_{23} + X_{24} &\leq 15986.4 \end{aligned} \quad (7.11)$$

The retailer related (demand) constraints can then be defined as:

$$\begin{aligned} X_{11} + X_{21} &\geq 6060.1 \\ X_{12} + X_{22} &\geq 7456.6 \\ X_{13} + X_{23} &\geq 5158.7 \\ X_{14} + X_{24} &\geq 5042 \end{aligned} \quad (7.12)$$

The carbon emissions and cost minimization problem then takes the form:

$$\begin{aligned} &Minimize(12.93 * X_{11} + 4.87 * X_{12} + 8.38 * X_{13} + 6.93 * X_{14} \\ &+ 10.54 * X_{21} + 11.75 * X_{22} + 14.02 * X_{23} + 10.87 * X_{24} + 0.0) \end{aligned}$$

subject to processor constraints (eq. 11)

subject to retailer constraints (eq. 12)

subject to, $0 \leq X_{ij}$

The simplified optimization problem with 6 rows, 8 columns and 16 elements is solved using the CBC optimizer. An optimal solution is found after 4 iterations with an objective value of 184,699.6500. With an output of 16,052 lb of beef from processor 1 and 7,667 lb of beef from processor 2, the final allocation of beef (decision variables) to be shipped to different retailers is shown in Table 7.3. X_{11} and X_{23} had the longest travel distance and consequently the highest carbon emissions, hence do not get selected.

Decision Variable	X_{11}	X_{12}	X_{13}	X_{14}
Beef Allocation (lb)	0.0	7457.0	5159.0	3436.0
Decision Variable	X_{21}	X_{22}	X_{23}	X_{24}
Beef Allocation (lb)	6061.0	0.0	0.0	1606.0

Table 7.3 Decision variable allocation for simplified 6 organization problem involving processor and retailers

A number of optimization problems are formulated for a bigger setup of organizations to demonstrate minimization of carbon emission costs between multiple source and destination pairs. Linear optimization problems are formulated between Breeder-Processor, Breeder-Distributor, Breeder-Retailer, Processor-Distributor, Processor-Retailer and Distributor - Retailer. Each problem consists of a supply matrix consisting of maximum amount of beef in pounds (lbs) that can be supplied from the destination and a demand matrix that represents the required amount of beef in pounds (lbs) at the source. The supply and demand matrix properties are reported for beef quantity in pounds (lbs). The carbon cost matrix properties are reported for carbon emissions in metric tonnes of CO_2eq . The objective value for optimization algorithm is reported in quantity of beef in pounds

(lbs) [194]. The total decision variables are the sum of assigned variables and the ones that are not assigned. All carbon costs are a result of the resource consumption (from reference framework) between each source and destination pair. Carbon emissions calculations also involve the resources consumed at the source but excludes destination. Carbon emissions between Breeder-Processor, Breeder-Distributor and Breeder-Retailer pairs are a result of the use of following resources:

$$C_{ij} = c_{energy} + c_{feed} + c_{byproducts} + c_{packaging} + c_{fertilizers} \\ c_{pesticides} + c_{processes} + c_{cleaners} + c_{machinery} \quad (7.13)$$

Carbon sequestration from plantation is only considered at the sources involving Breeder as the starting point. For simplicity, instead of counting live animals, the total amount of usable beef (63% of live cattle) in pounds leaving Breeder is considered in the supply matrix. Supply from breeder organization indicates animals that are ready to leave for processing (abattoir). A processor sink indicates amount of beef (carcass) that can be extracted from animals while a processor source indicates amount of beef that can be supplied in packaged form to the demanding organization.

A detailed summary of the results obtained at the optimizer node for a number of formulated optimization problems with different source-pair sets is summarized in Table 7.1 (and continuing till Table 7.2). The Breeder-Distributor pair used also involves resources consumed at the processor while a Breeder-Retailer pair also involves resources consumed at the processor and distributor organization. Similarly, a Processor-Retailer pair also involves resource consumption at the distributor organization. Carbon costs estimations do not involve the use of feed, fertilizers and pesticides when breeder organization is not involved. Carbon cost matrix values fall between [3000, 8000] CO_2eq with a uniform distribution for Breeder-Processor pair, between [5000, 12000] CO_2eq with a uniform distribution for Breeder-Distributor pair, between [4000, 9000] CO_2eq for Breeder-Retailer pair, between [3000, 7000] CO_2eq with uniform distribution for Processor-Distributor pair, between [2000, 7000] CO_2eq for Processor-Retailer pair and between [5, 15] CO_2eq for Distributor-Retailer pair. CO_2eq emissions between distributor and retailer are the least because it only involves fuel costs and cooling process for few days of transportation. Similarly, beef supply

and demand matrix values (in lbs) fall between different ranges in a uniform distribution for different source-destination pairs summarized in Table 7.4.

Source-Destination Pair	Source Destination Quantity Range (lbs)	Sink-Destination Quantity Range (lbs)
Breeder-Processor	(10000, 40000)	(5000, 13000)
Breeder-Distributor	(20000, 50000)	(7000, 15000)
Breeder-Retailer	(20000, 40000)	(10000, 15000)
Processor-Distributor	(30000, 40000)	(15000, 20000)
Processor-Retailer	(15000, 20000)	(5000, 10000)
Distributor-Retailer	(20000, 30000)	(10000, 15000)

Table 7.4 Beef range (in lb) used for different source-destination pairs utilized in the optimization problems defined in Table 7.1 and Table 7.2

7.4 Conclusion

The environmental impact of beef supply chain is substantial, contributing to accelerated environmental degradation. As beef supply chain become more complex with increasing number of participants, it becomes more difficult to share inter-organizational knowledge for joint supply chain optimization. To address the challenge of capturing detailed carbon emissions across the supply chain and using it for optimization of underlying resource consumption, we utilized our proposed decentralised collaboration framework, BeefMesh, and extended it to support optimization tasks. By focusing on the joint optimization of resource consumption and precise emission tracking, our application provided a flexible, comprehensive, and collaborative approach to recording, monitoring, and optimizing the carbon footprint across complex and fragmented supply chains, ultimately enhancing environmental management outcomes.

CHAPTER 8

ENABLING SECURE KNOWLEDGE TRANSFER PIPELINES USING BEEFMESH FRAMEWORK

Securing data pipelines in machine learning applications is crucial to maintaining data integrity, confidentiality, and privacy. This chapter provides an application use case that utilizes our implemented collaboration framework. Reliable machine learning data pipelines are configured with the deployment of blockchain channels and distributed databases. Utilization of blockchain in a highly decentralized architecture ensures immutable data storage and secure sharing among diverse stakeholders, addressing the limitations of traditional centralized systems. By incorporating federated learning with blockchain, we create a secure framework for data model consumption, management, and sharing. This method not only protects data privacy and security but also guarantees dependable and tamper-proof traceability for machine learning models throughout the supply chain.

8.1 Introduction

In the context of the beef supply chain, knowledge transfer is notably difficult due to the fragmentation of shared knowledge generating datasets, that are recorded and stored under diverse jurisdictions, each governed by varying privacy regulations (as summarized earlier in Chapter 2 and highlighted in Figure 1.3). In this chapter, we propose to improve and strengthen beef supply chain transparency and knowledge sharing capabilities using our federated collaboration framework. The framework coupled with secure machine learning model sharing pipelines, ensures reliable data models' storage, sharing, and aggregation while maintaining privacy and controlled user access.

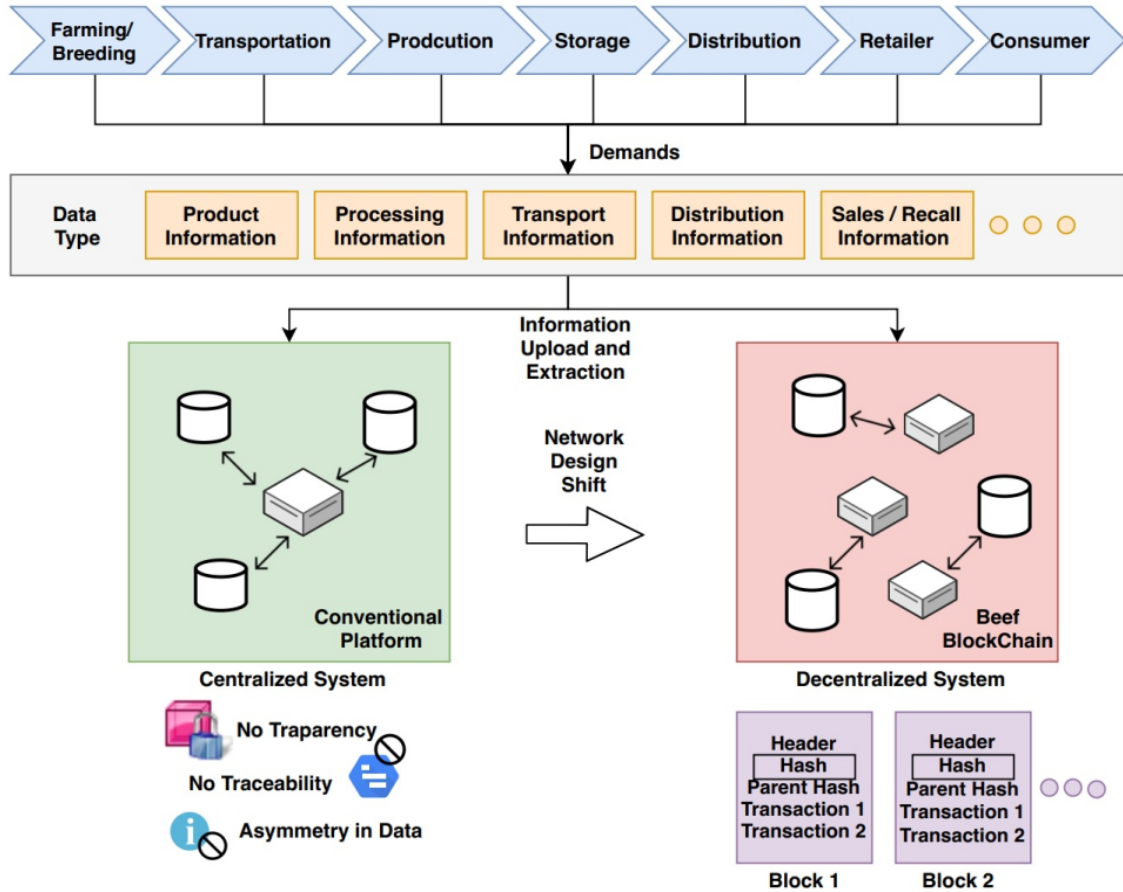
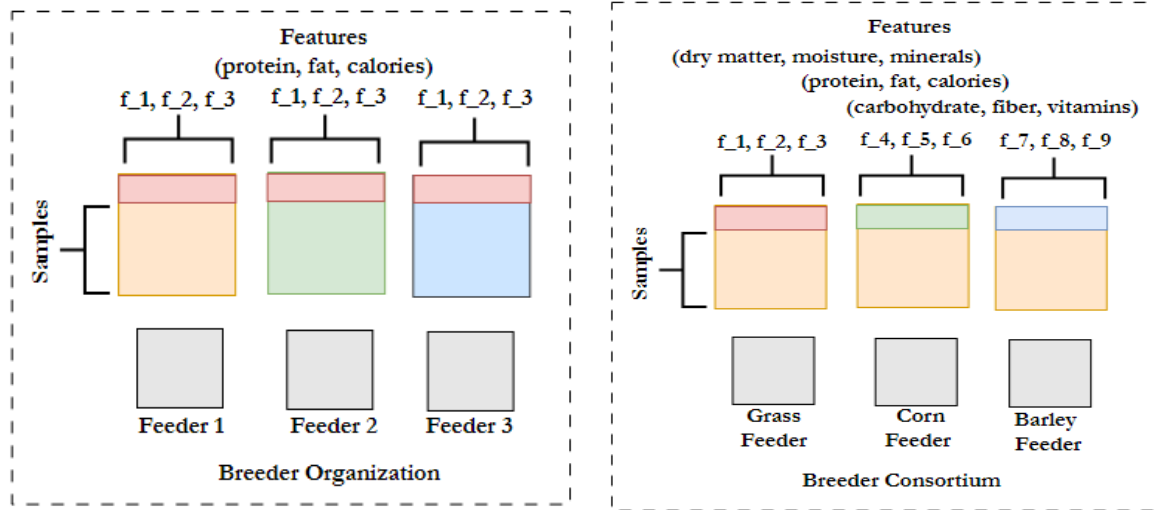


Figure 8.1 Moving from centralized to distributed and decentralized beef supply chain collaboration framework plays a key role in enabling secure federated learning data pipelines

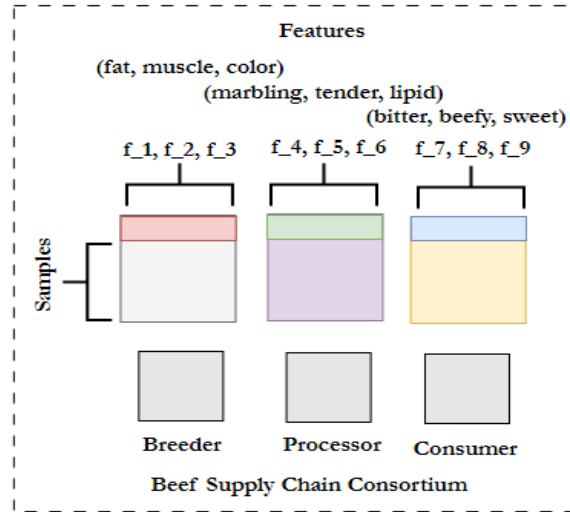
The collaboration framework allows configuring a connectivity and reliable data model sharing infrastructure with support for two distinct learning architectures, namely: (1) Federated Learning (FL) and (2) Collaborative Learning (CL), which are described in detail below.

Federated Learning (FL) is a machine learning method where a specific algorithm continually improves by individually training instances of itself in multiple independent sessions, utilizing segments of the dataset that are distributed across different databases [152]. The results obtained from these independent sessions of the FL algorithm are then combined using various techniques in a model aggregation server. The newly learned model is subsequently transmitted via secure server communication for use in the next iteration of training, involving segments of the dataset residing on local machines (as depicted in Figure 8.2). Mathematically, the objective function for the FL algorithm at the aggregation server can be defined as:



(a) Horizontal FL

(b) Vertical FL



(c) Transfer FL

Figure 8.2 Three different types of FL architectures can be leveraged using the proposed collaboration framework based on the variations of data (features and samples) in beef supply chain; namely (a) Horizontal FL (b) Vertical FL and (c) Transfer FL

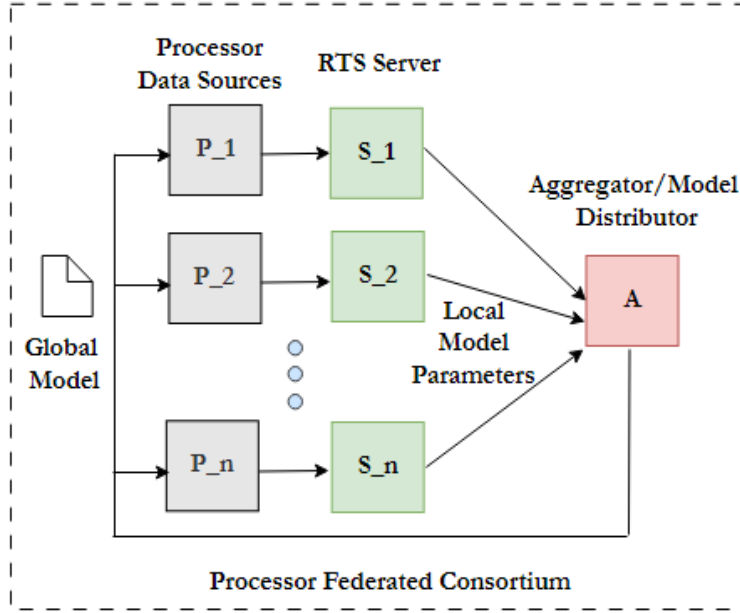


Figure 8.3 A new learned model in FL is formed by first collecting local models using secure Request to Send (RTS) server application and then combining the models at the aggregation server. The combined and optimized model is sent out to be used for next iteration at local nodes using IP communication

$$f(x_1, x_2, \dots, x_n) = \frac{1}{N} \sum_{i=1}^N f_i(x_i) \quad (8.1)$$

where N is total number of learning nodes, x_i is weight of the learned model on node i and f_i is the objective function utilized at local node i . In Federated Learning (FL), the objective function aims to iteratively optimize and enhance local objective functions by achieving consensus on the convergence of the parameter “x”. Several parameters play a pivotal role in shaping how the model evolves during training and aggregation with each iteration. Some of the parameters that can be tuned for FL include: (1) Total number of training rounds (K) (2) Total number of local nodes used for training (N) (3) Sample batch size at local nodes (S) used for training (4) The learning rate (r) of the algorithm. FL offers flexibility and can encompass various variations. These variations arise from adjusting the weights (W) of the underlying algorithm, adopting different loss (L) functions related to specific model weights, and varying the batch size of sample data.

Federated Learning (FL) can also be classified as a form of Distributed Learning (DL), as the training data is distributed across various nodes, and insights from all the data are aggregated

at a central location to gain a comprehensive understanding of the data's statistics. Distributed Learning (DL) relies on distributed resources, including databases and computational servers, which can either be federated or remain within the same private organization without the need for federation. In a typical scenario, local nodes in Distributed Learning (DL) are provided with an initial Machine Learning (ML) model from a central node, which is then executed on the local data. With each iteration, local model parameters are exchanged with the central server, amalgamated to create an updated global model, and subsequently transmitted back to the local node for local utilization (as illustrated in Figure 8.3).

In the proposed architecture, Federated Learning (FL) is facilitated through several means. Firstly, it is supported by integrating various types of devices, such as less capable but numerous mobile devices (referred to as “cross-device”), or more powerful client nodes (referred to as “cross-silo”). Secondly, FL leverages the variations or partitions in data for the learning process. Cross-device FL is typically carried out within an organization when all the devices are under the same organizational control (e.g., within a breeder organization). On the other hand, cross-silo FL involves nodes that fall under different organizational control (e.g., within a consortium of processors). From the perspective of data variations, three distinct categories of FL can be defined, which are: (1) Horizontal Federated Learning (Horizontal FL)(2) Vertical Federated Learning (Vertical FL) (3) Transfer Federated Learning (Transfer FL). These three types of FL configurations, as categorized by data variations, are illustrated in Figure 8.2.

In Horizontal Federated Learning (Horizontal FL), the features remain consistent across data sets located in different nodes, but the samples themselves may vary. For instance, when measuring cattle parameters related to feed, the features could include elements like “protein,” “fat,” and “calories.” However, the sample data, including both the size and specific values, may differ because cattle can move to different feeding locations within the same breeder organization.

In Vertical Federated Learning (Vertical FL), samples may be shared or have some overlap, but the features undergo change and do not overlap. For example, when a calf moves to fatteners, the feed type undergoes a complete transformation, as different types of grains and seeds are introduced.

Consequently, this results in the introduction of new features for machine learning. To illustrate, consider the case of Vertical FL within a consortium of breeders, each specializing in a different type of feed, such as grass-feed, corn-feed, and barley-feed.

As animals rotate among these three types of feeders over the weeks, entirely distinct features and corresponding sample data can be collected. For example, the grass-fed environment may focus on collecting samples for features like “dry matter,” “moisture,” and “minerals.” In contrast, the corn-fed environment may emphasize features such as “protein,” “fat,” and “calories.” Meanwhile, the barley-fed environment could be utilized to collect features like “carbohydrate,” “fiber,” and “calories” from the amount of feed provided to individual animals during a specific time period.

In the Transfer Federated Learning (Transfer FL), both the sample data and the features are either entirely different or have only partial overlap. For instance, when cattle move from breeders to processors, entirely new parameters (features) related to carcasses are appended to the cattle’s profile. Consider an organizational setup involving connected breeders, processors, and consumers attempting to perform federated Transfer learning. As animals traverse the supply chain, features like “fat,” “muscle,” and “color” are added to the animal’s profile at the breeder’s stage, while features like “marbling,” “tenderness,” and “lipid” are collected at the processor’s facility. Furthermore, features like “bitterness,” “beefiness,” and “sweetness” can be gathered from consumers and incorporated into the animal’s profile.

Both types of Federated Learning, whether based on device type or data type, can be broadly categorized under a “Model-Centric” Federated Learning. This categorization is due to the primary focus being on optimizing local and global models by exchanging them in each iteration. In contrast, a newer classification for Federated Learning employs a “Data-Centric” approach to learn from data. In the Data-Centric approach of Federated Learning, a central server is granted restricted access to data residing on a grid.

A ML application running on the central server scans through the data using privacy-preserving techniques, enabling the concealment of the actual data. Therefore, without revealing any part of the data, a data scientist can execute algorithms and utilize data on local machines to uncover

valuable insights. Privacy-preserving techniques that are mostly incorporated by default in current implementations of federated learning architectures include Differential Privacy (DP) and Private Set Intersection (PSI) [200, 201].

Considering the use of ML algorithm in FL, three different categories can be formed; namely (1) Federated Supervised Learning (FSL) (2) Federated Semi-Supervised Learning (FSSL) and (3) Federated Unsupervised Learning (FUL). Major ML techniques used in FSL include Linear Algorithms, Support Vector Machines (SVMs) and Decisions Trees. Major techniques used in FSSL include Federated Match algorithms and Logistic Regression. Major techniques used in FUL include Federated Generative Adversarial Network (GAN). Considering that the overall purpose of ML algorithm is to reduce the difference between predicted value (y) and real value (x) while learning parameter (w) with a rule (f), $f_w : x \rightarrow y$, a more formal mathematical definition for ML algorithm can be written as [202]:

$$\arg_w \min L(x, y, w) = ||f_w(x) - y|| \quad (8.2)$$

where L is the loss function that requires minimization with respect to the argument w . Extending it further to FL, we can write a generic ML formula for ML algorithm in Federated case:

$$\arg_w \min L(x, y, w) = \sum_n W_n L_n(x, y, w) \quad (8.3)$$

where n is the total number of federated clients, W_n is the weight of n_{th} client in a multi-client scenario consisting of decentralized users $\{U_1, U_2, \dots, U_n\}$ holding federated data $\{D_1, D_2, \dots, D_n\}$ respectively. Ideally, at the end of FL method, a Global Model M_{FED} is required that minimizes the loss accuracy between performance of federated global model on a test dataset Y_{FED} and performance of aggregated model on test dataset Y_{AGG} , i.e. $|T_{FED} - T_{AGG}| < \sigma$. In practice, a performance closer to this model is acceptable since it is not possible to obtain the aggregation model T_{AGG} because of privacy protection.

Unlike Federated Learning (FL), where parts of a dataset are distributed across various databases, Collaborative Learning (CL) within the framework relies on the integration of a central

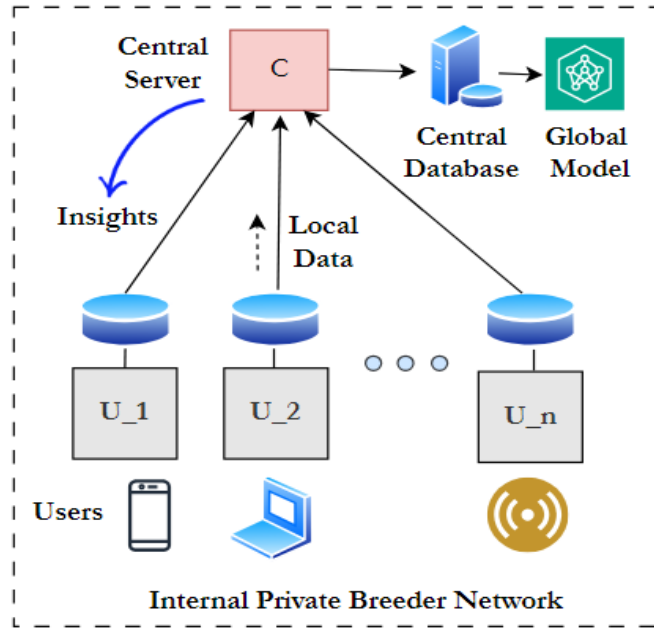


Figure 8.4 A CL system securely collects data from local nodes within an organization using private networking to run ML models at the central server

server. This central server securely collects data from local participating nodes and employs specialized Machine Learning (ML) algorithms for training. The central application server typically hosts a range of ML algorithms designed by an expert (data scientist). These algorithms are trained using the aggregated data, and the best results and insights among all the tested algorithms are shared individually with the local nodes upon completion.

8.2 The Federated Learning Data Pipelines Framework

The proposed collaboration framework (described in detail in Chapter 4 and Chapter 5) serve as a foundation for enabling various learning architectures within organizational setups through different database configurations and the integration of blockchain for storing critical and time-sensitive information. By integrating decentralized and distributed blockchain setup with distributed resources, our approach addresses challenges in knowledge transfer due to fragmented datasets and diverse privacy regulations (as highlighted in Figure 8.1). The collaboration framework, BeefMesh, as described in detail in Chapter 4, incorporates a hybrid permissioned consortium to record immutable transactions (as shown earlier in Figure 3.1, Figure 3.4 and Figure 3.6). The framework supports federated learning data pipelines by configuring independent data sharing blockchain

channels. The collaboration framework starts from the collaboration initiator and evolves into a fully decentralized infrastructure as described in detail in Chapter 4. The final form of the distributed and decentralised collaboration framework (partly shown in Figure 3.1), facilitates in private transactions related to federated learning, while accommodating various organizations and ensuring reliable data model keeping.

Centralized servers for federated learning in the framework can also be outsourced to 3rd party providers, often operating as cloud service applications. However, this approach comes with the drawback of granting access to sensitive user data. In the framework, CL coexists with nodes and a central server that fall under the same organizational entity, allowing sensitive data to be used for learning within a private network. The central ML server is more resourceful than the local nodes, as it not only hosts ML models but also performs computational tasks for Federated Learning.

An example of collaborative learning in a beef supply chain system involves nodes storing field sensor data related to cattle movement activities and nodes containing veterinary data for the same cattle. Both sets of records are transferred to a central server at the breeder facility to jointly calculate overall cattle health using ML algorithms. An illustration of such an architecture within the framework is presented in the provided diagram shown in Figure 8.4.

In our framework, we draw a clear distinction between Federated Learning (FL) and Federated Databases (FD). FL operates within the domain of ML and focuses on the iterative exchange of ML models among independently-owned databases belonging to different clients. On the other hand, FD represents a system of databases where a specific dataset is distributed across multiple databases but is perceived as a unified database by the client.

Within FD, an application running on a server processes client requests and seamlessly scans through all distributed databases, creating the illusion for the client that the requested data comes from a single, consolidated database. In FD, the server maintains a comprehensive record (catalog) of the entire dataset and how it is distributed across multiple databases. Therefore, an FD can be considered as a composite of more than one individual database.

The use of FD holds significant importance for supply chain participants for two primary

reasons. Firstly, cattle-related data is vast and continually expanding, and secondly, it is managed by multiple entities. For instance, cattle data at the same breeder location can have data added by veterinary personnel and incorporate movement data from field sensors.

In the proposed framework, FD is established for organizations by establishing secure connections to remote data servers (e.g., MongoDB) where the data is stored. This allows the retrieval of necessary data into a federated server, creating a unified view of data for the client, as illustrated earlier in Figure 4.5). Parameters that are used in the connection string to pull data from remote servers include *server name*, *login credentials*, connection parameters (*address* and *port number*) and *table details*. The format of a MYSQL connection string to remote server is given by:

schema://user[:pass]@host[:port]/db/table

where ‘schema’ is protocol for connection, ‘user’ and ‘pass’ are the login credentials, ‘host’ is the IP address of server, ‘db’ is the name of database and ‘table’ is the name of database table residing in the remote server.

8.3 Securing Federated Learning Data Flow Channels

Federated Learning (FL) was introduced to facilitate local data training without the necessity of transferring data to third-party servers. Nevertheless, FL continues to grapple with security and privacy concerns, as malevolent users or agents can disrupt the FL process through various means. These malicious attacks can be broadly categorized into three main groups [203].

Firstly, there is a risk of inadvertent data exposure or unauthorized data transfer to a third-party server or user who should not have access to the data. This can occur without the user’s consent or permission, particularly when attempting to run ML algorithms on a remote server and insecurely opening addresses and ports for connections.

Secondly, data privacy can be indirectly compromised if the ML model is exchanged without adequate generalization through insecure sharing connections.

Thirdly, the ML model itself can be corrupted if security precautions are not taken into account during the aggregation and sharing of the global model, as well as during the exchange of local

models between the server and clients. Consequently, potential targets for attacks by malicious users in an ML framework include manipulating data collection and data transmission.

The second and third ML model attacks can lead to corrupted training and testing. Since FL model files need to be circulated across different rounds, they are susceptible to manipulation. To enhance the security of FL model data during exchanges between the server and clients, blockchain channels are employed as a backup. Depending on the use case, distinct dedicated channels can be enabled. For example, three separate blockchain channels may be established to back up the sequence of improved model files for three different algorithms among a group of participants.

8.4 Example Applications and Discussion

To establish a seamless connectivity framework, the proposed system commences by initializing a blockchain-based federated learning channel. This channel serves as the starting point for launching the network. Organizations belonging to the consortium subsequently enter this channel and set up their respective permissioned network resources, each with predefined configurations and communication channels. The number of resources (such as peers and federated data nodes) that each organization joins with may vary depending on the specific scenario. Given the distributed nature of the system, participants have the flexibility to opt for new communication channels at any time, enabling them to engage in various federated learning applications without disrupting the initial ones.

A starting point for federated learning is that different sub-organizations in the beef supply chain network are independently owned. For example, farmers raise cattle independently from the breeder requirements. Processors are monitored and controlled by independent private companies that supply beef products to different distributors according to their financial and logistic requirements. Hence data is recorded without a global view of the information on blockchain. To illustrate federated learning applications utilizing our framework, we implement a number of applications that run over our proposed framework utilizing beef chain data to illustrate the usefulness and value of the system to different types of participants in the beef supply chain.

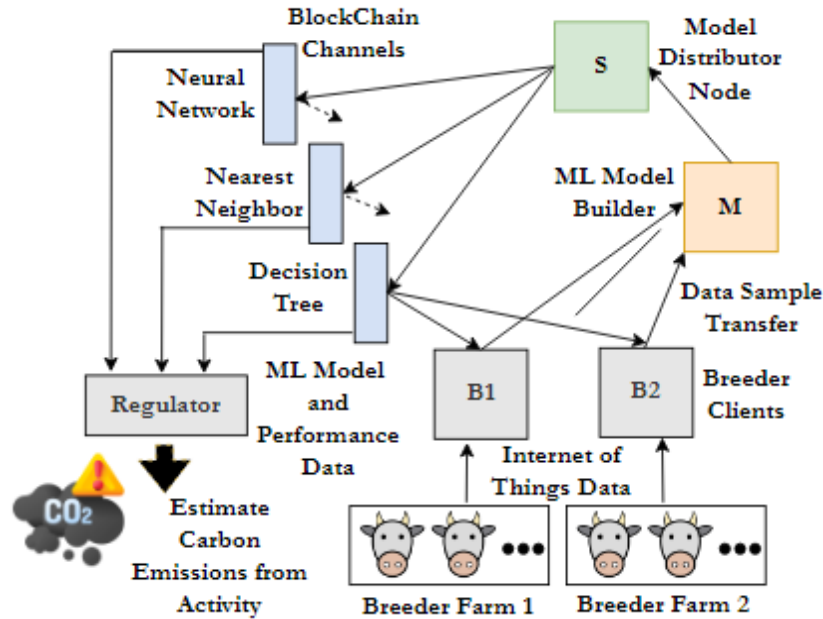


Figure 8.5 Network layout of the federated machine learning model building and performance evaluation in a breeder consortium utilizing multiple channels for different algorithms. In this example, animal related ‘urination activity’ is automatically detected and converted to carbon emissions against each organization

8.4.1 Breeder example case for automated emissions estimation from activity monitoring

An illustrative use case within a breeder consortium showcases the utilization of IoT data to automate the process of understanding animal behavior. This use case involves the collection of accelerometer data from animals, which is then stored in a relational database. Only a segment of the dataset is labeled, and the breeders collaboratively engage in the development of various machine learning models. These models are designed to be applied to future data, enabling the prediction of animal behavior.

The labeled data is transmitted to a node responsible for building and evaluating machine learning models. Subsequently, the models and their performance results are transferred to a distributor server. To ensure transparency and flexibility, the models and their performance metrics are shared using dedicated blockchain channels, as depicted in Figure 8.5. This approach allows breeders to make informed decisions about the utilization of specific models. If a breeder decides not to adopt certain types of models, they have the option to disconnect and will not have access to the underlying model details employed by other breeders in the future.

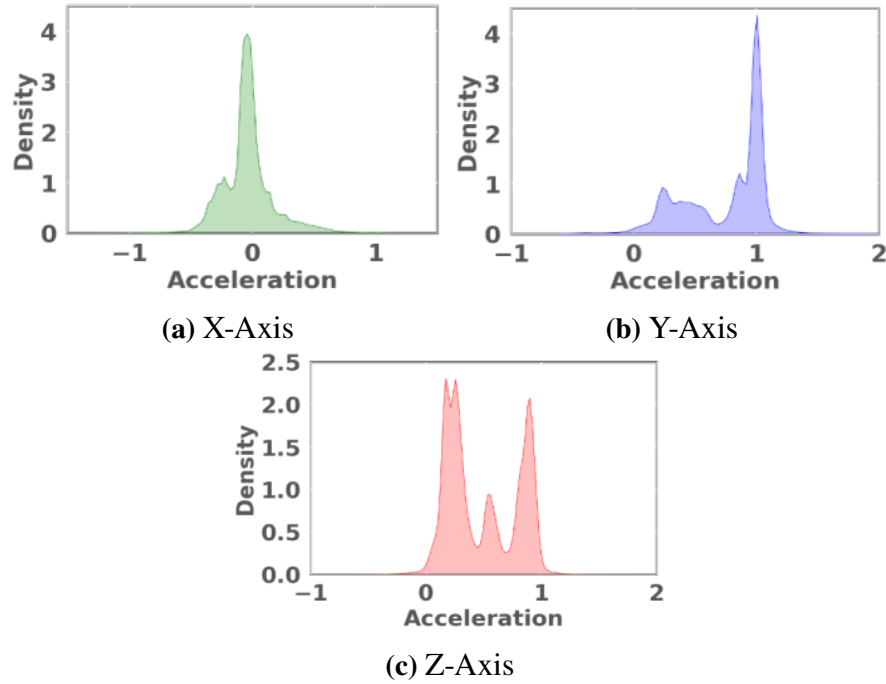


Figure 8.6 Acceleration distribution density for combined data at the ML model builder node in breeder federated consortium example

Animal	Cow1	Cow2	Cow3	Cow4	Cow5	Cow6
Data points	311876	269772	269903	224912	134904	179857

Table 8.1 Sample size of data used for breeder consortium example

For the carbon emissions calculation example at breeder consortium, a sample labelled dataset is taken from the [204]. The sample data consists of x,y and z values from an accelerometer samples at 25Hz (density shown in Figure 8.6). The complete dataset is comprised of samples from six cows. We assume cow1, cow2 and cow3 are with the first breeder client and cow4, cow5 and cow6 are with the second breeder client. Both clients send their labeled data stored in relational database to the model builder node. The sample data consisted of a total of 14 labels. The labels along with total data points in each category were: 1. Resting while standing (150130 samples) 2. Ruminating (53229 samples) 3. Moving (50199 samples) 4. Grazing (17613 samples) 5. Licking salt (10858 samples) 6. Feeding in stanchion (7934 samples) 7. Drinking (2476 samples) 8. Normal licking (1302 samples) 9. Resting while lying (764 samples) 10. Urinating (621 samples) 11. Attacking (366 samples) 12. Escaping (128 samples) 13. Being mounted (54 samples) 14. Other (554855

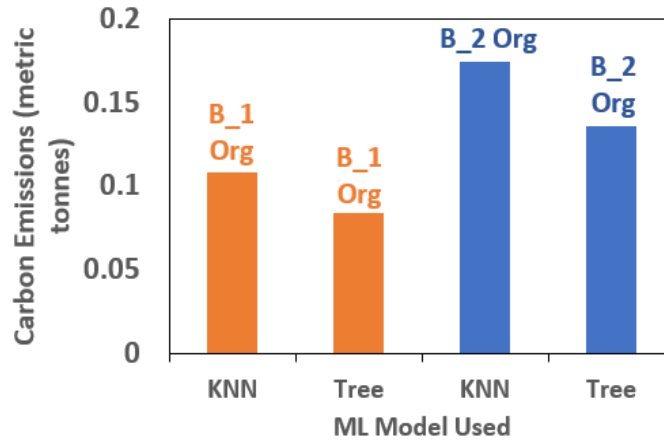


Figure 8.7 Emissions calculated at the regulatory authority from worst and best ML models using urination activity for breeder farm 1 (B_1 Org) and breeder farm 2 (B_2 Org)

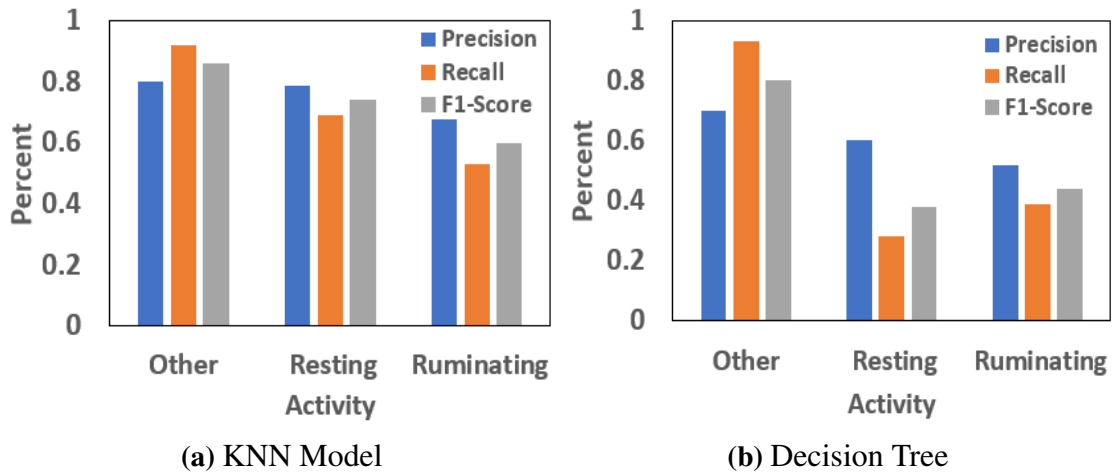


Figure 8.8 Precision, Recall and F1-Score for top three supporting activity classes in the breeder consortium example extracted from dedicated blockchain channels for KNN and decision tree

samples).

The decision to use the dataset [204], even though it is biased due to some subsets with more number of samples than other, is driven by several practical challenges. Firstly, there is a significant lack of comprehensive and realistic datasets for the beef supply chain. The industry's fragmented nature makes data collection difficult, and many companies are unwilling to share detailed information due to privacy and competitive concerns. In practice, the cost associated with gathering high-quality data from all parts of the supply chain under majority of scenarios is very high. Furthermore, real-world data often spans multiple jurisdictions, each with its own regulatory

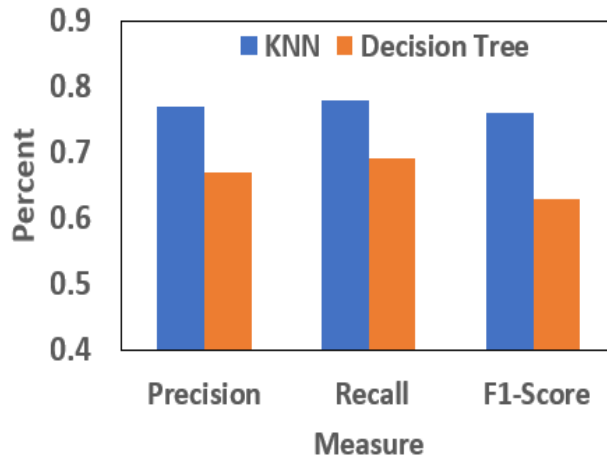


Figure 8.9 Weighted scores for Precision, Recall and F1-score for all activity classes measured by KNN and Decision Tree classifier models in breeder consortium example extracted from dedicated blockchain channels and calculated at the aggregator (model distributor) node

requirements, leading to inconsistencies and fragmentation. Excluding the heavily biased set would limit our ability to conduct meaningful research and analysis, as it would result in a lack of sufficient realistic data under real-world scenarios to draw valid conclusions. Despite its biases, the inclusion of the set still provides a decent foundation for analysis, enabling us to gain valuable insights and make informed decisions. Using this dataset allows us to reflect the complexities of the beef supply chain in terms of disproportionate datasets to a reasonable extent, thereby maintaining the study's overall effectiveness and relevance. Using the dataset, 3 different ML models were build including K Nearest Neighbors and Decision Tree to classify the activities and then shared with breeder clients along with performance metrics as shown in Figure 8.8(a) and Figure 8.8(b).

Mixed results for both breeders regarding Precision, Recall, and F-1 score parameters underscore the necessity of establishing distinct immutable blockchain channels for accessing previously more effective models. This is specially useful for the case when an inference needs to be made from an activity. For example, 'urination' activity can be converted to carbon emissions at the regulator side. Considering an animal produces 1.8 to 2.4 Liter (~3.5 Gallon a day) of urination per urination activity period, a mean of 2.1 Liter can be taken for reference [205]. Cattle urine mostly contains Nitrogen concentration which is more potent than CO_2 . The (N) concentration can be in the forms of Nitrous Oxide (N_2O), Ammonia (NH_3), Di-Nitrogen (N_2) and Nitrate (NO_3). Considering

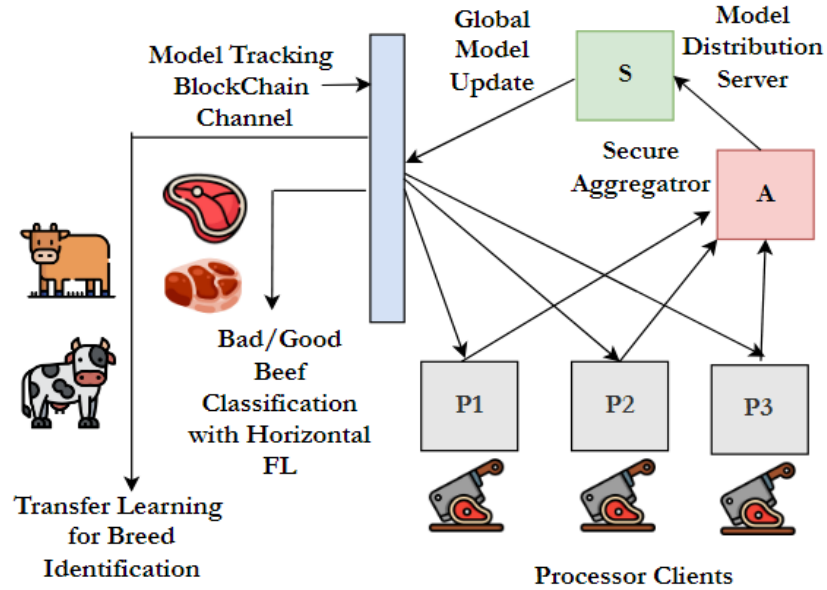


Figure 8.10 Network layout of the secure federated ML model training example for processor consortium within beef chain utilizing permitted blockchain channel

major component of emissions being N_2O and each liter of urine producing 3 to 20 grams ($\sim 12\text{g/L}$) of N_2O [206], each successfully detected urination activity can be converted to equivalent carbon emissions. The success of detecting carbon emissions in this case therefore depends upon using the most accurately predicting model from the stored models on blockchain. The difference of emissions calculations for worst and best models for the two breeder organizations is shown in Figure 8.7. N_2O in grams are first converted to CO_2 and then to CO_2eq in metric tonnes. Breeder_1 (B1) organization holds cattle set {cow1,cow4,cow6} while Breeder_2 (B2) organization holds cattle set {cow1,cow4,cow6}. The worst case model was Decision Tree with average prediction accuracy of 0.57 and the best case model was KNN with average prediction accuracy of 0.71 for urination activity.

8.4.2 Processor example case for cattle type recognition with transfer learning

A use case example of transfer federated learning using blockchain framework for model transfer is presented. A consortium of two processors mutually collaborate to recognize a set of 5 different breeds of cattle, namely Ayrshire, brown, holstein, jersey and red dane so that they can be sorted out before being processed (as shown in Figure 8.11). The dataset for the example is

taken from [207] where each category contains more than 200 samples. The two processors (P_1) and (P_2) organizations collectively use 160 random samples for each category divided into 70/30 for training and testing with random sampling at each epoch. A pre-trained network of ConvNet (from ImageNet) is used with fine tuning to extract features while training on the given cattle breed dataset. ImageNet is not a specific Convolutional Neural Network (ConvNet) itself but rather a large-scale dataset containing millions of labeled images across thousands of categories. We use the Pytorch library and use the ConvNet model for learning. ConvNet is a deep learning architecture specially designed for processing grid-like data, such as images and video frames.

We first initialize the learning network (ConvNet) at the aggregator for the two processors and normalize it for the dataset. The datasets are first resized and normalized to a 3-dimensional NP-array (grid of values indexed by tuple of positive values) and trained for 10 epochs using linear regression based decay measure and Stochastic Gradient Descent optimizer with cross entropy based loss measure for each epoch. Fitness of the model is measured at each epoch using training and validation loss. Once trained, the model is exchanged via blockchain channel with another processor that has new samples of same 5 breeds not used in training or testing. The transferred model is used to predict directly the 5 categories of cattle with 40 samples in each case. The results of training for the first two processors and prediction with model transfer for the third processor is shown in Figure 8.12 and Figure 8.13 with Holstein breed being recognized with the highest prediction accuracy using transferred learning model.

In transfer learning, it is often observed that the training loss exceeds the validation loss due to several key factors. Firstly, during training, regularization methods such as dropout and weight regularization introduce noise to prevent over-fitting, thereby increasing the training loss. Batch normalization also behaves differently between training and validation phases, using batch-specific statistics during training, which adds variability. Use of early stopping and check-pointing technique in transfer learning, which allows saving model states that minimize validation loss, can sometimes result in a lower validation loss compared to training loss. Data augmentation, applied to training data, adds further variability, making it more challenging to achieve low training loss. Learning

rate schedules also cause temporary increases in training loss. Additionally, in federated transfer learning, where datasets from different sources represent the same animals, differences in the datasets and the presence of unseen data during training can contribute to higher training loss. Despite these discrepancies, the model's accuracy is still effective, as the validation loss remains a reliable indicator of the model's performance on unseen data. Thus, the model retains its ability to generalize well and provide valuable insights despite the higher training loss.

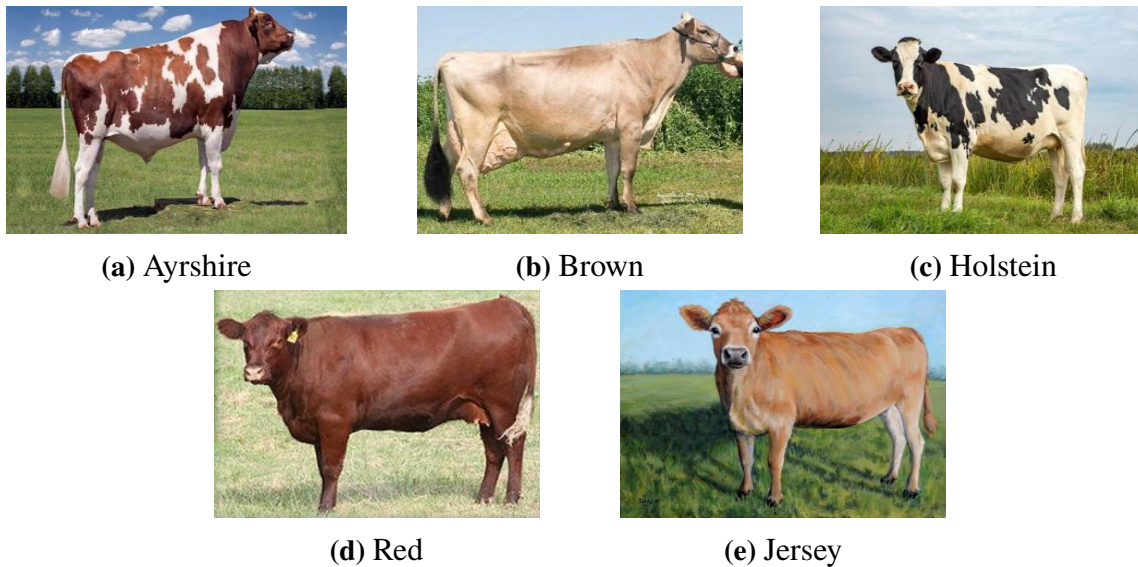


Figure 8.11 Cattle breeds used for transfer learning example in a processor consortium setup

8.4.3 Processor example case for automated beef quality detection using horizontal ensemble learning

An example application is presented for image based beef quality assessment. Three processing plants collaborate to learn a ML model that can predict that state of the beef (good, bad) (as shown in Figure 8.15) by looking at images (total 948 images taken from [208]) to automate the process of discarding low quality or bad beef cuts. The processing plants don't want to expose or send their data to external entities, so they only send out 5% of their sample images to a dedicated node that securely aggregates the model, updates it and then sends it to a model distributor agent (server) to send the updated model back to processor clients. The three processor clients share a common blockchain channel with the global model distribution server node to keep track of the model



Figure 8.12 Training and Validation loss for transfer learning (breed type) at the aggregator for processor consortium organization

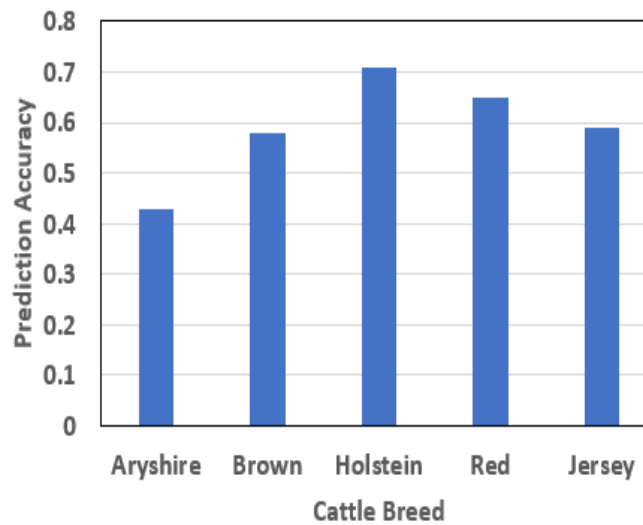


Figure 8.13 Breed type prediction with model transfer for processor_3 in processor consortium example

updates as shown in Figure 8.10. This setup constitutes a horizontal machine learning framework as shown in Figure 8.2(a).

For our example horizontal machine learning processor case, each node runs the updated model on their training data in each round where each round consists of one epoch. The model is built using a sequence of 3 convolution layers with pooling followed by a dense fully connected layer and a softmax layer. A global model is created by using the average of model weights from the

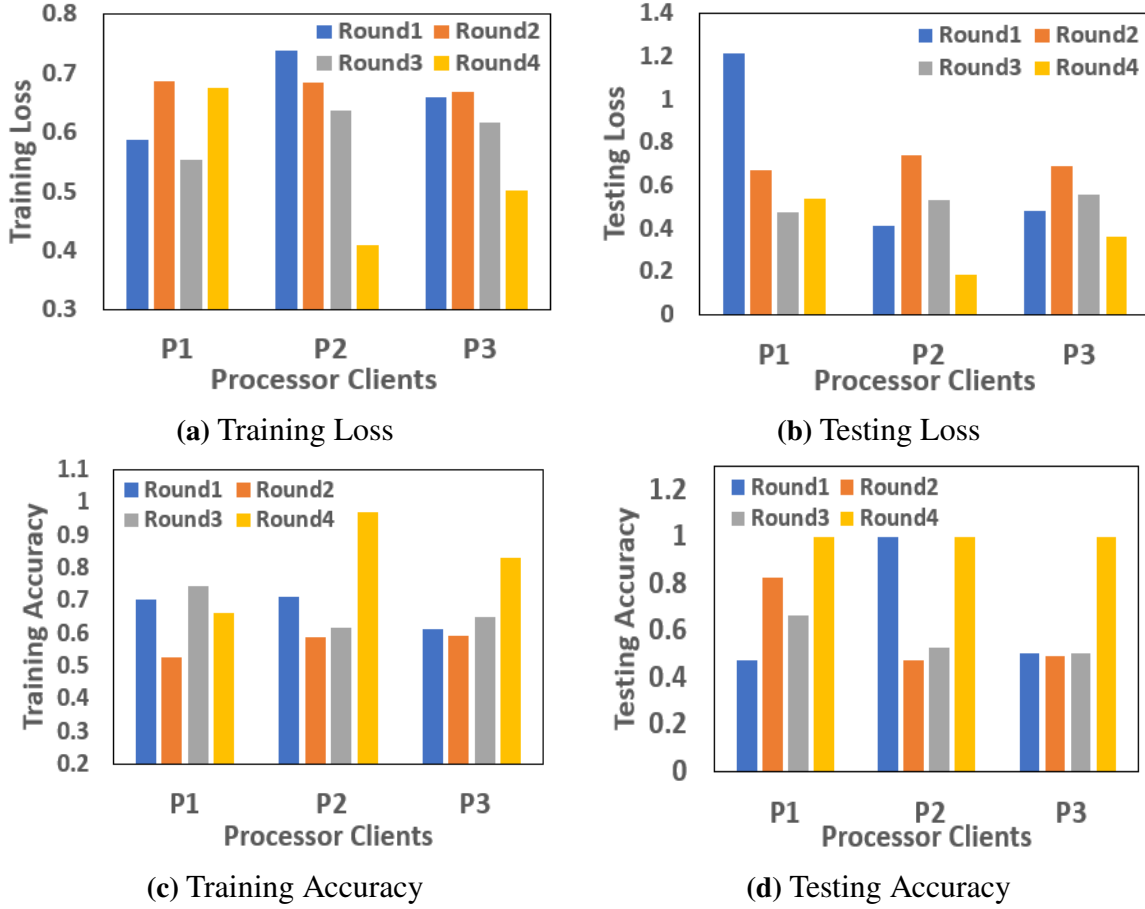
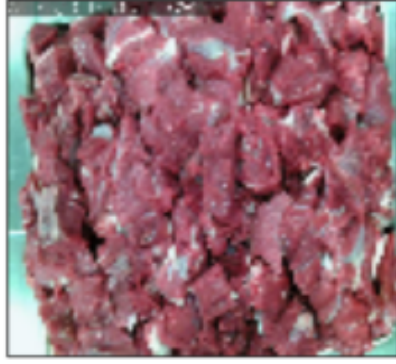


Figure 8.14 Training and testing accuracy and errors for federated machine learning stored and fetched from private federated blockchain channel in processor consortium example for horizontal learning

three client models and then using it for the weights of the global model. Since the global improved model is shared over blockchain, any of the previous models can be reused if any processor client notices better results for their side. Further any use of malicious data from any of the clients to create bias in the model can be tracked by comparing with previous version and tracking the exact round. The network layout of the secure federated machine learning learning model training is shown in Figure 8.10. Training and testing accuracy and error results (as shown in Figure 8.14) for federated machine learning example are stored and fetched from private federated blockchain channel for the processor consortium example. Due to random distribution of images across all three processors, mixed results for training and testing loss and accuracy can be seen, depicting the need to maintain immutable traceable model version overtime on blockchain channel to retrieve a



(a) Fresh Beef



(b) Bad Beef

Figure 8.15 Example image of (left) fresh and (right) bad beef samples from the training samples used for horizontal learning

better version of model when required.

In the horizontal federated learning scenario (as shown in Figure 8.14), the testing loss is observed to be lower than the training loss. This occurs due to several reasons. Firstly, during training, dropout and weight regularization are utilized to prevent over-fitting, which adds extra noise and increases the training loss. Batch normalization also contributes to this effect due to the batch-specific statistics during training. Batch-specific statistics directly adds variability. Additionally, in the context of horizontal federated learning, random processor data is distributed across multiple sources. Random data with unknown and widely changing statistics in multiple processor sources can result in training data being more diverse and noisy in relation to the random testing data. In horizontal learning, the unseen data is partitioned randomly, which can result in testing sets that are more representative and less noisy than the training sets. Early stopping and check-pointing based on validation loss also leads to lower testing loss, as the model parameters are optimized to minimize loss on unseen data. As more rounds of training and testing take place, more data is filtered by ML model, and the algorithm encounters greater variations, leading to better results for training and testing loss. But since our target here is to achieve federated learning by utilizing the collaborative framework with secure model sharing data pipelines, we focus more on achieving effective and visible accuracy in fewer rounds. Since data is distributed randomly with varying statistics on different processor nodes, the federated learning model may not

increase monotonically during initial rounds. The testing accuracy, however, increases over time in fewer rounds, suggesting that the model generalizes well to new, unseen data. Nevertheless, the distributed collaborative model sharing framework allows reliable storage of different versions of the ML model files. This helps different processor participants in choosing the ML models for their organizational use, that does not over-fit with their localized data and one that performs reliably in real-world scenarios with unseen random data. In addition, the use of blockchain channels for sharing and storing ML models in different rounds of the learning process does not add considerable amount of time. As shown in Table 5.7, read and write tasks on blockchain channels take around 0.1 sec each for files that are directly stored in string, YAML or in JSON formats. For larger ML files that are shared using CID from IPFS storage, it takes an average of 5 sec for writing and reading files of around 200MB with combined calls for both blockchain and IPFS services. Since reading and writing CID data on blockchain takes around 0.2 sec (as show in Table 5.7), the additional time to write and read from IPFS nodes can be estimated from Table 5.5.

Another type of federated learning can easily be leveraged from the existing connectivity framework called Hierarchical Federated Learning which is an advanced machine learning approach designed to address the challenges of privacy, scalability, and efficient model training in distributed systems. This technique extends the principles of Federated Learning by introducing a hierarchical structure to the participating devices or nodes, allowing for more organized and efficient model aggregation.

In a Hierarchical Federated Learning system, participating devices are organized into a hierarchical structure. This hierarchy typically consists of multiple levels, with each level representing a group of nodes. Higher levels may represent more powerful or central nodes, such as data centers or cloud servers, while lower levels may comprise edge devices or user devices. At each level of the hierarchy, devices perform local model training on their own data without sharing sensitive information. This local training allows devices to compute model updates based on their local datasets while maintaining data privacy. Model updates are shared and aggregated within each level of the hierarchy. The aggregation can occur in a hierarchical manner, where higher-level nodes

collect and aggregate updates from lower-level nodes. This hierarchical aggregation minimizes the amount of data that needs to be transmitted between levels, reducing communication overhead. A global model, representing the collective knowledge of all devices in the system, is constructed through the aggregation of model updates from the hierarchical structure. This global model is then shared back down the hierarchy, allowing lower-level nodes to benefit from the insights gained at higher levels.

Next, we demonstrate a horizontal federated learning example case for determining beef quality through images. We particularly use the FedML library and tools to enable this setup [209]. The example case uses two processors (P_1 and P_2) which serve as silo (or clients) and each one uses a GPU (Graphics Processing Unit) instance. P_1 trains the good and bad beef class images on 2 nodes with 1 GPU for each process and P_2 trains its independent model using a single GPU on a single node. The example provided here demonstrates a scenario where there are two silos/clients, and each of them has access to multiple GPUs. Silo-1 trains the model on 2 nodes, with each node having 1 GPU, while Silo-2 trains its model using 1 GPU on a single node. Model is trained using FedAvg optimizer, 10 epochs for 30 communication rounds. Each client uses a Stochastic Gradient Descent (SGD) optimizer locally with a learning rate of 0.001 and weight decay of 0.001. Validation frequency for testing is set to 5 rounds while MQTT protocol format is used as the backend communication platform. Finally the global models (in .pt format) are stored and shared with other nodes using blockchain channels. With a random 50/50 distribution of beef quality images among the two processor nodes P_1 and P_2 nodes and using the FedML horizontal setup of federated learning, a training loss of 2.23401 and an accuracy of 0.912 is achieved.

8.5 Conclusion

Ensuring secure data pipelines in machine learning is essential for preserving data integrity, confidentiality, and privacy. Our approach utilized blockchain technology and secure communication channels to establish a decentralized framework that allows for immutable data storage and secure sharing among various stakeholders. This framework effectively addresses the limitations of traditional centralized systems. By integrating federated learning with blockchain, we demon-

strated a robust method for data consumption, extraction, and knowledge transfer. This not only enhanced data privacy and security but also provided reliable and tamper-proof traceability across the supply chain. We illustrated the effectiveness of our framework using examples of collaborating organizations, including processors and breeders. These collaborations showcased the practical application of our secure data pipeline in real-world machine learning scenarios, highlighting improvements in trust, transparency, and cooperation among different entities. The decentralized and distributed approach for collaboration in federated learning scenarios, ensures that critical information such as the machine learning models, are securely managed and shared, ultimately benefiting all participants in the supply chain.

CHAPTER 9

CONCLUSION AND FUTURE DIRECTIONS

The beef supply chain network is a complex system that incorporates various subsystems such as cattle breeding, stock management, feedlot operations, cold transportation, packaging, distribution, retail operations and waste management. Each subsystem, typically managed by distinct private organizations, processes the same product, (the beef), but shares limited information with other subsystems. The restricted scrutiny of subsystems by regulators and fragmentation among organizations, leads to a lack of communication, reduced trust, less transparency, and compromised traceability. Current technologies, which rely on private and centralized ledgers, primarily around point-of-sale connectivity, cannot be used for sharing of critical information that generates common knowledge, especially during emergencies like outbreaks. With the input of one subsystem, heavily dependent on the output of another subsystem, the fragmented nature of beef supply chain represents a missed opportunity to collaborate, share traceability data and leverage machine learning algorithms with federated data sources for extracting and sharing common supply chain knowledge.

To overcome the issues of fragmentation and limited connectivity in supply chains, particularly beef chain, we proposed and demonstrated a decentralized and distributed collaboration framework, BeefMesh, coupled with blockchain infrastructure and distributed resources. We leveraged decentralized resources and distributed methodologies to demonstrate the process of extracting data and learning common information in a non-pervasive way from federated data sources. The use of permissioned connectivity channels between supply chain participants setup by a collaborative

process, provisions secure data consumption, data management, information extraction, and knowledge transfer without losing control of data by a contributing client. The federated permissioned framework ensures the timely dissemination of critical information, especially during emergencies like outbreak, thereby strengthening trust, transparency, traceability, and collaboration.

9.1 Summary of the Thesis

In Chapter 1 of the thesis, we discussed the fragmentation and complexity of the current supply chain network, highlighting the resulting communication challenges. We emphasized the need for a collaborative framework to address issues of privacy, data control, and flexible collaboration, aiming to create a cohesive and efficient supply chain, particularly for the beef industry. In Chapter 2, we examined the importance of collaboration in enhancing supply chain efficiency and integration, highlighting the application of theoretical frameworks and modeling for optimization. We discussed regulatory compliance for food safety, sector-specific challenges in the beef supply chain, and the transformative potential of blockchain for traceability and data integrity. This chapter underscores the necessity of strategic collaboration, regulatory adherence, advanced technologies, and effective modeling to build resilient and transparent supply chains. In Chapter 3, we presented a supply chain collaboration framework that integrates blockchain, databases, and IoT sensors to facilitate projects like traceability and tracking. This versatile and scalable system ensures participant control over components and data, incorporates stringent access controls, and uses a permissioned consortium for security. The framework supports dynamic, decentralized collaboration, ensuring data integrity, security, and participant autonomy, making it adaptable to modern supply chain needs. In Chapter 4, we detailed the implementation of the proposed collaboration framework with description of the software and application tools used.

In Chapter 5, we demonstrated a traceability application for cattle in beef supply chain network, utilizing the BeefMesh collaboration framework. In Chapter 6, we addressed the environmental impacts of the beef supply chain, focusing on the GHG emissions. We used our decentralized blockchain-based framework, BeefMesh, and integrated it with IoTs and databases to track detailed emissions data throughout the supply chain. This framework incorporated diverse information

sources, demonstrating its capability to securely capture data, communicate policies, and facilitate reliable traceability and flexible environmental data sharing. This highlighted the frameworks capability for promoting emissions reduction and management in complex food supply chains. In Chapter 7, we extended BeefMesh to develop an application that optimizes resource consumption that directly contributes to GHG emissions, enhancing environmental management outcomes. Finally, in Chapter 8, we extended the BeefMesh framework using blockchain and federated learning to ensure secure data pipelines for machine learning model sharing, enhancing data integrity, confidentiality, and privacy among supply chain participants.

9.2 Supported Applications and Future Directions

The future of the supply chains lies in harnessing the potential of blockchain and other decentralized and distributed platforms to address traceability, food safety, efficiency, sustainability, and fraud prevention. A list of the potential applications that can be directly or indirectly implemented leveraging the proposed decentralized and distributed collaboration framework include:

1. Enhancing traceability and transparency in fragmented supply chains.
2. Measuring environmental impact from supply chain functions and outputs.
3. Facilitating and safeguarding the data flow of shared knowledge related to collaborative organizational activities.
4. Ensuring food safety by tracking product related activities in the supply chain.
5. Ensuring animal welfare by monitoring cattle handling at different stages.
6. Tracking supply chain inefficiency by monitoring and coordinating costs, delays, and output waste.
7. Countering fraud, counterfeit products and market manipulation by collaboratively enforcing regulatory compliance.

8. Balancing market access and trade monopoly by allowing formation of regional collaborative organizational groups.
9. Neutralizing labor issues by provisioning collaborative platforms to share policies related to labor disputes and work conditions.
10. Managing disease control by tracking and identifying the root cause of an outbreak in the supply chain.

With further enhancements of the proposed collaboration framework, it is possible to implement and support some other important applications. These applications can potentially help organizations by working together to remove any forms of material scarcity, uncontrolled price hikes, unavailability of freights, traffic congestion while forecasting market demands, and incorporating consumer response for enhanced transparency of supply chain functions. In future, we plan to continue improving the framework by incorporating AI based applications into the system, e.g., processing data from chain of events pertaining to various organizational activities and mapping it into meaningful automated actions.

9.3 Limitations of the Proposed Framework

While the proposed collaboration framework offers significant benefits, it is important to acknowledge the following limitations:

1. Services within the framework are configured with securely exposed API's for client use, alongside user interface for certain applications. However, some applications may lack a fully developed frontend.
2. The use of blockchain channels as a data pipeline for sharing and storing data models and other information (e.g., control data) within the federated architecture enhances the reliability of the data-sharing process. Federated learning architectures inherently utilize built-in data privacy techniques, secure aggregation, data encryption, and access control mechanisms to ensure the security of the learning process.

3. Due to the inclusion of multiple containerized applications, users might need to manually adjust ports for certain services, depending on the availability of open ports on their host machines.
4. The framework was developed and tested in a Linux environment (Ubuntu 22.04) and therefore, may not function properly on OS with different scripting formats and execution styles.
5. The performance of the distributed framework, enabled by an overlay network, is constrained by the actual performance of the underlying physical network.
6. Organizations are limited in their blockchain functionality until a proper Orderer node with a consensus role is configured and operational. Without an Orderer node, clients send channel-related change requests to other organizations with a properly configured Orderer node before they can start using the channel.
7. The use of a shared network drive utilizing the GlusterFS application requires at least two dedicated server nodes.
8. Some of the guidelines provided for securing the framework may only be applicable to Linux OS and could differ for other systems.
9. The carbon emissions-related application utilizes factors available in the literature, some of which are based on assumptions and have limitations within the LCA process.

Attributions and Other Contributions

Content from chapters of this work, in part or in full, is under preparation for submission (or has been submitted), that are authored by Salman Ali, Cedric Gondro, Qiben Yan and Wolfgang Banzhaf. Ali and Gondro contributed on the original idea. Ali contributed in implementation of software, conducting experiments and writing the original manuscript. Yan and Banzhaf, contributed on improving the idea and on the writing and revision of the manuscripts. Owen helped with improving the thesis write-up.

BIBLIOGRAPHY

- [1] Matthias Meier and Eugenio Pinto. COVID-19 supply chain disruptions. *Covid Economics*, 48(1):139–170, 2020.
- [2] Sanjoy Kumar Paul, Priyabrata Chowdhury, Md Abdul Moktadir, and Kwok Hung Lau. Supply chain recovery challenges in the wake of COVID-19 pandemic. *Journal of Business Research*, 136:316–329, 2021.
- [3] Dabo Guan, Daoping Wang, Stephane Hallegatte, Steven J Davis, Jingwen Huo, Shuping Li, Yangchun Bai, Tianyang Lei, Qianyu Xue, D’Maris Coffman, et al. Global supply-chain effects of COVID-19 control measures. *Nature Human Behaviour*, 4(6):577–587, 2020.
- [4] Abel Yeboah-Ofori and Shareeful Islam. Cyber security threat modeling for supply chain organizational environments. *Future Internet*, 11(3):63, 2019.
- [5] Javid Moosavi, Amir M Fathollahi-Fard, and Maxim A Dulebenets. Supply chain disruption during the COVID-19 pandemic: Recognizing potential disruption management strategies. *International Journal of Disaster Risk Reduction*, 75:102983, 2022.
- [6] Sara Quach, Park Thaichon, Kelly D Martin, Scott Weaven, and Robert W Palmatier. Digital technologies: tensions in privacy and data. *Journal of the Academy of Marketing Science*, 50(6):1299–1323, 2022.
- [7] Ashish Kumar Jha, Maher AN Agi, and Eric WT Ngai. A note on big data analytics capability development in supply chain. *Decision Support Systems*, 138:113382, 2020.
- [8] Progressive Publishing. Global beef supply statistics, 2023. Available at: <https://www.progressivepublish.com/downloads/2023/general/2023-pc-stats-highres.pdf> (Accessed: 2024-03-04).
- [9] CSC Yip, W Lam, and R Fielding. A summary of meat intakes and health burdens. *European Journal of Clinical Nutrition*, 72(1):18–29, 2018.
- [10] Michael Dent. Plant-based and cultured meat 2020-2030: Technologies, markets and forecasts in novel meat replacements. *IDTechEx*, pages 1–286, 2020. Technical Report.
- [11] Petra Vidergar, Matjaž Perc, and Rebeka Kovačič Lukman. A survey of the life cycle assessment of food supply chains. *Journal of Cleaner Production*, 286:125506, 2021.
- [12] Mehmet Soysal, Jacqueline M Bloemhof-Ruwaard, and Jack GAJ Van Der Vorst. Modelling food logistics networks with emission considerations: The case of an international beef supply chain. *International Journal of Production Economics*, 152:57–70, 2014.
- [13] Douglas M Lambert and Martha C Cooper. Issues in supply chain management. *Industrial*

Marketing Management, 29(1):65–83, 2000.

- [14] Guanyi Lu, Xenophon Koufteros, Srinivas Talluri, and G Tomas M Hult. Deployment of supply chain security practices: antecedents and consequences. *Decision Sciences*, 50(3):459–497, 2019.
- [15] Deepak Arunachalam, Niraj Kumar, and John Paul Kawalek. Understanding big data analytics capabilities in supply chain management: Unravelling the issues, challenges and implications for practice. *Transportation Research Part E: Logistics and Transportation Review*, 114:416–436, 2018.
- [16] James S Drouillard. Current situation and future trends for beef production in the United States of America—A review. *Asian-Australasian Journal of Animal Sciences*, 31(7):1007–1016, 2018.
- [17] Muzammil Hussain, Waheed Javed, Owais Hakeem, Abdullah Yousafzai, Alisha Younas, Mazhar Javed Awan, Haitham Nobanee, and Azlan Mohd Zain. Blockchain-based IoT devices in supply chain management: a systematic literature review. *Sustainability*, 13(24):13646, 2021.
- [18] Prince Waqas Khan, Yung-Cheol Byun, and Namje Park. IoT-blockchain enabled optimized provenance system for food industry 4.0 using advanced deep learning. *Sensors*, 20(10):2990, 2020.
- [19] Aaron M Shew, Heather A Snell, Rodolfo M Nayga Jr, and Mary C Lacity. Consumer valuation of blockchain traceability for beef in the United States. *Applied Economic Perspectives and Policy*, 44(1):299–323, 2022.
- [20] Tanvir Ferdousi, Don Gruenbacher, and Caterina M Scoglio. A permissioned distributed ledger for the US beef cattle supply chain. *IEEE Access*, 8:154833–154847, 2020.
- [21] Pankaj Dutta, Tsan-Ming Choi, Surabhi Somani, and Richa Butala. Blockchain technology in supply chain operations: Applications, challenges and research opportunities. *Transportation Research Part E: Logistics and Transportation Review*, 142:102067, 2020.
- [22] Udit Agarwal, Vinay Rishiwal, Sudeep Tanwar, Rashmi Chaudhary, Gulshan Sharma, Pitshou N Bokoro, and Ravi Sharma. Blockchain technology for secure supply chain management: A comprehensive review. *IEEE Access*, 10:85493–85517, 2022.
- [23] Jayasree Sengupta, Sushmita Ruj, and Sipra Das Bit. A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT. *Journal of Network and Computer Applications*, 149:102481, 2020.
- [24] Sana Al-Farsi, Muhammad Mazhar Rathore, and Spiros Bakiras. Security of blockchain-based supply chain management systems: Challenges and opportunities. *Applied Sciences*,

11(12):5585, 2021.

- [25] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 107:841–853, 2020.
- [26] Feng Tian. A supply chain traceability system for food safety based on HACCP, blockchain & Internet of Things. In *2017 International Conference on Service Systems and Service Management*, pages 1–6. IEEE, 2017.
- [27] Shoufeng Cao, Warwick Powell, Marcus Foth, Valeri Natanelov, Thomas Miller, and Uwe Dulleck. Strengthening consumer trust in beef supply chain traceability with a blockchain-based human-machine reconcile mechanism. *Computers and Electronics in Agriculture*, 180:105886, 2021.
- [28] Kentaroh Toyoda, P Takis Mathiopoulos, Iwao Sasase, and Tomoaki Ohtsuki. A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain. *IEEE Access*, 5:17465–17477, 2017.
- [29] Hokey Min. Blockchain technology for enhancing supply chain resilience. *Business Horizons*, 62(1):35–45, 2019.
- [30] Yong Yuan and Fei-Yue Wang. Towards blockchain-based intelligent transportation systems. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2663–2668. IEEE, 2016.
- [31] Haya R Hasan and Khaled Salah. Blockchain-based proof of delivery of physical assets with single and multiple transporters. *IEEE Access*, 6:46781–46793, 2018.
- [32] Houtian Ge, Miguel Gómez, and Christian Peters. Modeling and optimizing the beef supply chain in the northeastern us. *Agricultural Economics*, 53(5):702–718, 2022.
- [33] Thanos E Goltsos, Aris A Syntetos, Christoph H Glock, and George Ioannou. Inventory–forecasting: Mind the gap. *European Journal of Operational Research*, 299(2):397–419, 2022.
- [34] Paul H. Zipkin. *Foundations of Inventory Management*. McGraw-Hill, 2000.
- [35] Sven Axssäter. *Inventory control*, volume 225. Springer, 2015.
- [36] James P. Womack, Daniel T. Jones, and Daniel Roos. *The Machine That Changed the World*. Free Press, 1990.
- [37] Edward Allen Silver, David F Pyke, Rein Peterson, et al. *Inventory management and production planning and scheduling*, volume 3. Wiley New York, 1998.

- [38] Kalyan T Talluri and Garrett J Van Ryzin. *The theory and practice of revenue management*, volume 68. Springer Science & Business Media, 2006.
- [39] Julien Bramel and David Simchi-Levi. *The logic of logistics: theory, algorithms and applications for logistics management*. Springer Berlin, 1998.
- [40] Parisa Alizadeh, Hosein Mohammadi, Naser Shahnoushi, Sayed Saghaian, and Alireza Pooya. Application of system thinking approach in identifying the challenges of beef value chain. *AGRIS On-Line Papers in Economics and Informatics*, 12(2):3–16, 2020.
- [41] T Wise and Betsy Rakocy. Hogging the gains from trade. *The real winners from US trade and agricultural policies. GDAE Policy Brief*, pages 10–01, 2010.
- [42] Kees-Jan van Dorp. Beef labelling: The emergence of transparency. *Supply Chain Management: An International Journal*, 8(1):32–40, 2003.
- [43] Kelsey Robson, Moira Dean, Stephanie Brooks, Simon Haughey, and Christopher Elliott. A 20-year analysis of reported food fraud in the global beef supply chain. *Food Control*, 116:107310, 2020.
- [44] John Lynch and Raymond Pierrehumbert. Climate impacts of cultured meat and beef cattle. *Frontiers in Sustainable Food Systems*, 3:421–491, 2019.
- [45] Juan F Galvez, JC Mejuto, and J Simal-Gandara. Future challenges on the use of blockchain for food traceability analysis. *TrAC Trends in Analytical Chemistry*, 107:222–232, 2018.
- [46] Shengnan Sun, Xinping Wang, and Yan Zhang. Sustainable traceability in the food supply chain: The impact of consumer willingness to pay. *Sustainability*, 9(6):999, 2017.
- [47] Petter Olsen and Melania Borit. The components of a food traceability system. *Trends in Food Science & Technology*, 77:143–149, 2018.
- [48] Alan S Kolok, Jonathan M Ali, Eleanor G Rogan, and Shannon L Bartelt-Hunt. The fate of synthetic and endogenous hormones used in the us beef and dairy industries and the potential for human exposure. *Current Environmental Health Reports*, 5(2):225–232, 2018.
- [49] GD Snowder, L Dale Van Vleck, LV Cundiff, and GL Bennett. Bovine respiratory disease in feedlot cattle: environmental, genetic, and economic factors. *Journal of Animal Science*, 84(8):1999–2008, 2006.
- [50] Michael Boland and Ted Schroeder. Marginal value of quality attributes for natural and organic beef. *Journal of Agricultural and Applied Economics*, 34(1):39–49, 2002.
- [51] Daniel Nepstad, David McGrath, Claudia Stickler, Ane Alencar, Andrea Azevedo, Briana Swette, Tathiana Bezerra, Maria DiGiano, João Shimada, Ronaldo Seroa da Motta, et al.

- Slowing amazon deforestation through public policy and interventions in beef and soy supply chains. *Science*, 344(6188):1118–1123, 2014.
- [52] Victoria Salin. Information technology and cattle-beef supply chains. *American Journal of Agricultural Economics*, 82(5):1105–1111, 2000.
 - [53] Christine G Elsik, Deepak R Unni, Colin M Diesh, Aditi Tayal, Marianne L Emery, Hung N Nguyen, and Darren E Hagen. Bovine genome database: new tools for gleaning function from the bos taurus genome. *Nucleic Acids Research*, 44(D1):D834–D839, 2016.
 - [54] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. Big data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, 30(4):431–448, 2018.
 - [55] John G. Keogh, Abderahman Rejeb, Nida Khan, Kevin Dean, and Karen J. Hand. Blockchain and GS1 standards in the food chain: A review of the possibilities and challenges. *Trends in Food Science & Technology*, 103:171–181, 2020.
 - [56] Céline Faverjon, Abraham Bernstein, Rolf Grütter, Christina Nathues, Heiko Nathues, Cristina Sarasua, Martin Sterchi, Maria-Elena Vargas, and John Berezowski. A transdisciplinary approach supporting the implementation of a big data project in livestock production: An example from the swiss pig production industry. *Frontiers in Veterinary Science*, 6:215, 2019.
 - [57] Frederick Winslow Taylor. *The principles of scientific management*. Harper & Brothers, 1919.
 - [58] Safaa Sindi, Michael Roe, Safaa Sindi, and Michael Roe. The evolution of supply chains and logistics. *Strategic supply chain management: the development of a diagnostic model*, pages 7–25, 2017.
 - [59] Donald J Bowersox and David J Closs. *Logistical Management: The integrated supply chain process*. McGraw-Hill, 1996.
 - [60] Angappa Gunasekaran and Eric WT Ngai. Information systems in supply chain integration and management. *European journal of operational research*, 159(2):269–295, 2004.
 - [61] Mani Subramani. How do suppliers benefit from information technology use in supply chain relationships? *MIS Quarterly*, pages 45–73, 2004.
 - [62] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business & Information Systems Engineering*, 6:239–242, 2014.
 - [63] Erik Hofmann and Marco Rüsch. Industry 4.0 and the current status as well as future prospects on logistics. *Computers in Industry*, 89:23–34, 2017.

- [64] Rommert Dekker, Jacqueline Bloemhof, and Ioannis Mallidis. Operations research for green logistics—an overview of aspects, issues, contributions and challenges. *European Journal of Operational Research*, 219(3):671–679, 2012.
- [65] Dmitry Ivanov. Supply chain viability and the covid-19 pandemic: a conceptual and formal generalisation of four major adaptation strategies. *International Journal of Production Research*, 59(12):3535–3552, 2021.
- [66] A Ravi Ravindran, Donald P Warsing Jr, and Paul M Griffin. *Supply chain engineering: Models and applications*. CRC Press, 2023.
- [67] Eldon Glen Caldwell Marin. Toward smart manufacturing and supply chain logistics. *IEEE Technology and Engineering Management Society Body of Knowledge (TEMSBOK)*, pages 147–166, 2023.
- [68] Marc Levinson. *The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger*. Princeton University Press, 2006.
- [69] Martin Campbell-Kelly, William F Aspray, Jeffrey R Yost, Honghong Tinn, and Gerardo Con Díaz. *Computer: A history of the information machine*. Routledge, 2023.
- [70] Jack PC Kleijnen and PJ Rens. Impact revisited: A critical analysis of IBM’s inventory package “IMPACT”. *Production and Inventory Management, Journal of the American Production and Inventory Control Society*, 19(1):71–90, 1978.
- [71] Margaret A Emmelhainz. *EDI: Total Management Guide*. John Wiley & Sons, Inc., 1992.
- [72] Barry M Leiner, Vinton G Cerf, David D Clark, Robert E Kahn, Leonard Kleinrock, Daniel C Lynch, Jon Postel, Larry G Roberts, and Stephen Wolff. A brief history of the internet. *ACM SIGCOMM computer communication review*, 39(5):22–31, 2009.
- [73] Mehmet Gümüş and James H Bookbinder. Cross-docking and its implications in location-distribution systems. *Journal of Business logistics*, 25(2):199–228, 2004.
- [74] James P Womack, Daniel T Jones, and Daniel Roos. *The machine that changed the world: The story of lean production—Toyota’s secret weapon in the global car wars that is now revolutionizing world industry*. Simon and Schuster, 2007.
- [75] Matthias Holweg. The genealogy of lean production. *Journal of Operations Management*, 25(2):420–437, 2007.
- [76] Dan Masse. Rfid handbook: Fundamentals and applications in contactless smart cards and identification second edition. *Microwave Journal*, 47(10):168–169, 2004.
- [77] Dara G Schniederjans, Carla Curado, and Mehrnaz Khalajhedayati. Supply chain digitisation

- trends: An integration of knowledge management. *International Journal of Production Economics*, 220:107439, 2020.
- [78] Umang Soni, Vipul Jain, and Sameer Kumar. Measuring supply chain resilience using a deterministic modeling approach. *Computers & Industrial Engineering*, 74:11–25, 2014.
 - [79] J George Shanthikumar, David D Yao, and W Henk M Zijm. *Stochastic modeling and optimization of manufacturing systems and supply chains*, volume 63. Springer Science & Business Media, 2003.
 - [80] Haralambos Sarimveis, Panagiotis Patrinos, Chris D Tarantilis, and Chris T Kiranoudis. Dynamic modeling and control of supply chain systems: A review. *Computers & Operations Research*, 35(11):3530–3561, 2008.
 - [81] J Blackhurst, Teresa Wu, and P O’grady. Network-based approach to modelling uncertainty in a supply chain. *International Journal of Production Research*, 42(8):1639–1658, 2004.
 - [82] Angappa Gunasekaran, Nachiappan Subramanian, and Thanos Papadopoulos. Information technology for competitive advantage within logistics and supply chains: A review. *Transportation Research Part E: Logistics and Transportation Review*, 99:14–33, 2017.
 - [83] John M Antle. Benefits and costs of food safety regulation. *Food Policy*, 24(6):605–623, 1999.
 - [84] Barbara A Almanza and Melissa S Nesmith. Food safety certification regulations in the united states. *Journal of Environmental Health*, 66(9):10–14, 2004.
 - [85] Jianrong Zhang and Tejas Bhatt. A guidance document on the best practices in food traceability. *Comprehensive Reviews in Food Science and Food Safety*, 13(5):1074–1103, 2014.
 - [86] Lina Kantiani, Marta Llorca, Josep Sanchís, Marinella Farré, and Damià Barceló. Emerging food contaminants: A review. *Analytical and Bioanalytical Chemistry*, 398(6):2413–2427, 2010.
 - [87] Katie Stewart and Lawrence O. Gostin. Food and drug administration regulation of food safety. *JAMA*, 306(1):88–89, 2011.
 - [88] European Food Safety Authority. Use of the EFSA comprehensive european food consumption database in exposure assessment. *EFSA Journal*, 9(3):2097, 2011.
 - [89] Glynn T Tonsor and Ted C Schroeder. Livestock identification: Lessons for the us beef industry from the australian system. *Journal of International Food & Agribusiness Marketing*, 18(3-4):103–118, 2006.
 - [90] JL Jouve. Principles of food safety legislation. *Food Control*, 9(2-3):75–81, 1998.

- [91] Kamal Souali, Othmane Rahmaoui, and Mohammed Ouzzif. An overview of traceability: Definitions and techniques. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pages 789–793. IEEE, 2016.
- [92] GS1. GS1 global traceability standard, June 2017. Available at: https://www.gs1.org/sites/default/files/docs/traceability/GS1_Global_Traceability_Standard_i2.pdf (Accessed: 2023-07-28).
- [93] US Food and Drug Administration. FSMA final rule for preventive controls for human food. Current Good Manufacturing Practice, Hazard Analysis, and Risk-Based Preventive Controls for Human Food, 2016. Available at: <https://www.fda.gov/food/food-safety-modernization-act-fsma/fsma-final-rule-preventive-controls-human-food> (Accessed: 2023-04-10).
- [94] ISO. ISO 22005: 2007 traceability in the feed and food chain—general principles and basic requirements for system design and implementation. 2007.
- [95] Bowen Tan, Jiaqi Yan, Si Chen, and Xingchen Liu. The impact of blockchain on food supply chain: the case of Walmart. In *International Conference on Smart Blockchain*, pages 167–177. Springer, 2018.
- [96] Montserrat Espiñeira and Francisco J Santaclara. *Advances in food traceability techniques and technologies: improving quality throughout the food chain*. Woodhead Publishing, 2016.
- [97] Dan W. Shike. Beef cattle feed efficiency. *Driftless Region Beef Conference Proceedings*, pages 1–10, 2013.
- [98] RM Dixon, C Playford, and DB Coates. Nutrition of beef breeder cows in the dry tropics. 2. effects of time of weaning and diet quality on breeder performance. *Animal Production Science*, 51(6):529–540, 2011.
- [99] JA Archer, EC Richardson, RM Herd, and PF Arthur. Potential for selection to improve efficiency of feed use in beef cattle: A review. *Australian Journal of Agricultural Research*, 50(2):147–162, 1999.
- [100] ISO Technical Committee. Traceability in the feed and food chain—general principles and basic requirements for system design and implementation. International Organization for Standardization (ISO), Standard No. ISO 22005:2007, pp. 1-12, July 2007. Available at: <https://www.iso.org/standard/36297.html> (Accessed: 2022-10-04).
- [101] Angelo Corallo, Roberto Paiano, Anna Lisa Guido, Andrea Pandurino, Maria Elena Latino, and Marta Menegoli. Intelligent monitoring internet of things based system for agri-food value chain traceability and transparency: A framework proposed. In *2018 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, pages 1–6. IEEE, 2018.

- [102] Affaf Shahid, Ahmad Almogren, Nadeem Javaid, Fahad Ahmad Al-Zahrani, Mansour Zuair, and Masoom Alam. Blockchain-based agri-food supply chain: A complete solution. *IEEE Access*, 8:69230–69243, 2020.
- [103] Filippo Gandino, Bartolomeo Montrucchio, Maurizio Rebaudengo, and Erwing R Sanchez. On improving automation by integrating RFID in the traceability management of the agri-food sector. *IEEE Transactions on Industrial Electronics*, 56(7):2357–2365, 2009.
- [104] Corrado Costa, Francesca Antonucci, Federico Pallottino, Jacopo Aguzzi, David Sarriá, and Paolo Menesatti. A review on agri-food supply chain traceability by means of RFID technology. *Food and Bioprocess Technology*, 6(2):353–366, 2013.
- [105] Leena Kumari, K Narsaiah, MK Grewal, and RK Anurag. Application of RFID in agri-food sector. *Trends in Food Science & Technology*, 43(2):144–161, 2015.
- [106] Zhang Yiying, Ruan Yuanlong, Liu Fei, Shang Jing, and Liu Song. Research on meat food traceability system based on RFID technology. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 2172–2175. IEEE, 2019.
- [107] Pengcheng Nie, Yong He, Na Wu, and Hui Zhang. Agricultural products traceability system applications. In *Agricultural Internet of Things*, pages 373–400. Springer, 2021.
- [108] Magnus Kempe, Carolina Sachs, and H. Skoog. Blockchain use cases for food traceability and control. Kairos Future, November 2017. Available at: <https://www.kairosfuture.com/publications/reports/blockchain-use-cases-for-food-tracking-and-control/> (Accessed: 2023-11-08).
- [109] Gildas Avoine and Philippe Oechslin. RFID traceability: A multilayer problem. In *International Conference on Financial Cryptography and Data Security*, pages 125–140. Springer, 2005.
- [110] Atul Kumar, Ankit Kumar Jain, and Mohit Dua. A comprehensive taxonomy of security and privacy issues in RFID. *Complex & Intelligent Systems*, 7(3):1327–1347, 2021.
- [111] Laslo Tarjan, Ivana Šenk, Srdjan Tegeltija, Stevan Stankovski, and Gordana Ostojic. A readability analysis for QR code application in a traceability system. *Computers and Electronics in Agriculture*, 109:1–11, 2014.
- [112] Wang Xueyuan and Yang Bo. Research and design of traceability system of agricultural products. In *2018 International Conference on Engineering Simulation and Intelligent Control (ESAIC)*, pages 384–388. IEEE, 2018.
- [113] Hong Mei Gao. Study on the application of the QRcode technology in the farm product supply chain traceability system. In *Applied Mechanics and Materials*, volume 321, pages

- 3056–3060. Trans Tech Publ, 2013.
- [114] Danny Pigini and Massimo Conti. NFC-based traceability in the food chain. *Sustainability*, 9(10):1910, 2017.
 - [115] A Sankara Narayanan. QR codes and security solutions. *International Journal of Computer Science and Telecommunications*, 3(7):69–72, 2012.
 - [116] Weigbin Hong, Yefan Cai, Ziru Yu, and Xiangyang Yu. An agri-product traceability system based on iot and blockchain technology. In *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, pages 254–255. IEEE, 2018.
 - [117] Wei Zhou and Selwyn Piramuthu. IoT and supply chain traceability. In *International Conference on Future Network Systems and Security*, pages 156–165. Springer, 2015.
 - [118] Luca Catarinucci, Inigo Cuinas, Isabel Exposito, Riccardo Colella, Jose Antonio Gay Fernandez, and Luciano Tarricone. RFID and WSNs for traceability of agricultural goods from farm to fork: electromagnetic and deployment aspects on wine test-cases. In *SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks*, pages 1–4. IEEE, 2011.
 - [119] Ricardo Badia-Melis, Puneet Mishra, and Luis Ruiz-García. Food traceability: New trends and recent advances. A review. *Food Control*, 57:393–401, 2015.
 - [120] Daesik Ko, Yunsik Kwak, and Seokil Song. Real time traceability and monitoring system for agricultural products based on wireless sensor network. *International Journal of Distributed Sensor Networks*, 10(6):832510, 2014.
 - [121] Hua Wang, Zonghua Zhang, and Tarek Taleb. Special issue on security and privacy of IoT. *World Wide Web*, 21(1):1–6, 2018.
 - [122] Ajay Jangra, Richa, Swati, and Priyanka. Wireless Sensor Network (WSN): Architectural design issues and challenges. *International Journal on Computer Science and Engineering*, 2(9):3089–3094, 2010.
 - [123] KN Pankov. Testing, verification and validation of distributed ledger systems. In *2020 Systems of Signals Generating and Processing in the Field of on Board Communications*, pages 1–9. IEEE, 2020.
 - [124] Michael Nofer, Peter Gomber, Oliver Hinz, and Dirk Schiereck. Blockchain. *Business & Information Systems Engineering*, 59(3):183–187, 2017.
 - [125] Joe Zou, Zhongli Dong, Allen Shao, Peng Zhuang, Wei Li, and Albert Y Zomaya. 3D-DAG: A high performance DAG network with eventual consistency and finality. In *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, pages 262–263.

IEEE, 2018.

- [126] Leemon Baird. Hashgraph consensus: Fair, fast, byzantine fault tolerance, May 2016. Available at: <https://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf> (Accessed: 2022-05-11).
- [127] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 557–564. IEEE, 2017.
- [128] Anderson Domeneguetto Felipe and Antonio Carlos Demanboro. Smart contracts and blockchain: An application model for traceability in the beef supply chain. In *Brazilian Technology Symposium*, pages 499–508. Springer, 2019.
- [129] Wen Lin, David L Ortega, Danielle Ufer, Vincenzina Caputo, and Titus Awokuse. Blockchain-based traceability and demand for US beef in China. *Applied Economic Perspectives and Policy*, 44(1):253–272, 2022.
- [130] Myo Min Aung and Yoon Seok Chang. Traceability in a food supply chain: Safety and quality perspectives. *Food Control*, 39:172–184, 2014.
- [131] Qinghua Lu and Xiwei Xu. Adaptable blockchain-based systems: A case study for product traceability. *Ieee Software*, 34(6):21–27, 2017.
- [132] Abderahman Rejeb, John G Keogh, and Horst Treiblmaier. Leveraging the internet of things and blockchain technology in supply chain management. *Future Internet*, 11(7):161, 2019.
- [133] Saikat Mondal, Kanishka P Wijewardena, Saranraj Karuppuswami, Nitya Kriti, Deepak Kumar, and Premjeet Chahal. Blockchain inspired RFID-based information architecture for food supply chain. *IEEE Internet of Things Journal*, 6(3):5803–5813, 2019.
- [134] Huawei Huang, Jianru Lin, Baichuan Zheng, Zibin Zheng, and Jing Bian. When blockchain meets distributed file systems: An overview, challenges, and open issues. *IEEE Access*, 8:50574–50586, 2020.
- [135] Gavina Baralla, Andrea Pinna, Roberto Tonelli, Michele Marchesi, and Simona Ibba. Ensuring transparency and traceability of food local products: A blockchain application to a smart tourism region. *Concurrency and Computation: Practice and Experience*, 33(1):e5857, 2021.
- [136] Haihui Huang, Xiuxiu Zhou, and Jun Liu. Food supply chain traceability scheme based on blockchain and EPC technology. In *International Conference on Smart Blockchain*, pages 32–42. Springer, 2019.
- [137] Kaijun Leng, Ya Bi, Linbo Jing, Han-Chi Fu, and Inneke Van Nieuwenhuysse. Research

- on agricultural supply chain system with double chain architecture based on blockchain technology. *Future Generation Computer Systems*, 86:641–649, 2018.
- [138] Khaled Salah, Nishara Nizamuddin, Raja Jayaraman, and Mohammad Omar. Blockchain-based soybean traceability in agricultural supply chain. *IEEE Access*, 7:73295–73305, 2019.
 - [139] Reshma Kamath. Food traceability on blockchain: Walmart’s pork and mango pilots with ibm. *The Journal of the British Blockchain Association*, 1(1):3712, 2018.
 - [140] Juan Benet. IPFS - content addressed, versioned, P2P file system. *arXiv preprint arXiv:1407.3561*, July 2014. DRAFT 3.
 - [141] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger Fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, pages 1–15, 2018.
 - [142] Liudmila Zavolokina, Rafael Ziolkowski, Ingrid Bauer, and Gerhard Schwabe. Management, governance, and value creation in a blockchain consortium. *MIS Quarterly Executive*, 19(1):1–17, 2020.
 - [143] Docker, Inc. Docker swarm documentation, 2024. Available at: <https://docs.docker.com/engine/swarm/> (Accessed: 2024-04-10).
 - [144] Gluster Community. GlusterFS, 2024. Available at: <https://github.com/gluster/glusterfs> (Accessed: 2024-02-12).
 - [145] Mainflux. Mainflux - Open source IoT platform, 2024. Available at: <https://github.com/mainflux/mainflux> (Accessed: 2023-07-26).
 - [146] The Prometheus Authors. Prometheus: The monitoring toolkit, 2024. Available at: <https://github.com/prometheus/prometheus> (Accessed: 2024-01-05).
 - [147] Grafana Labs. Grafana, 2024. Available at: <https://github.com/grafana/grafana> (Accessed: 2024-05-12).
 - [148] Yuxi Li, Liang Qiao, and Zhihan Lv. An optimized byzantine fault tolerance algorithm for consortium blockchain. *Peer-to-Peer Networking and Applications*, 14:2826–2839, 2021.
 - [149] Carlisle Adams, Patrick Cain, Denis Pinkas, and Robert Zuccherato. Internet X. 509 public key infrastructure time-stamp protocol (TSP). Request for Comments RFC 3161, Internet Engineering Task Force (IETF), August 2001. Standards Track.
 - [150] Amazon Web Services. ETL vs ELT - difference between data-processing approaches. AWS, 2023. Available at: <https://aws.amazon.com/compare/the-difference-between-etl-and-elt/>

(Accessed: 2023-12-26).

- [151] Dejan Mijić and Ervin Varga. Unified IoT platform architecture platforms as major IoT building blocks. In *2018 International Conference on Computing and Network Communications (CoCoNet)*, pages 6–13. IEEE, 2018.
- [152] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.
- [153] Shashank Kumar, Rakesh D Raut, Kirti Nayal, Sascha Kraus, Vinay Surendra Yadav, and Balkrishna E Narkhede. To identify industry 4.0 and circular economy adoption barriers in the agriculture supply chain by using ISM-ANP. *Journal of Cleaner Production*, 293:126023, 2021.
- [154] Simone Tagliapietra and Georg Zachmann. The role of carbon pricing in decarbonizing supply chains. *Energy Policy*, 145:111727, 2021.
- [155] ZOU Caineng, Bo Xiong, XUE Huaqing, Dewen ZHENG, GE Zhixin, WANG Ying, Luyang JIANG, PAN Songqi, and WU Songtao. The role of new energy in carbon neutral. *Petroleum Exploration and Development*, 48(2):480–491, 2021.
- [156] Divya Pandey, Madhoolika Agrawal, and Jai Shanker Pandey. Carbon footprint: Current methods of estimation. *Environmental Monitoring and Assessment*, 178(1):135–160, 2011.
- [157] Zhu Liu, Zhu Deng, Steven J Davis, Clement Giron, and Philippe Ciais. Monitoring global carbon emissions in 2021. *Nature Reviews Earth & Environment*, 3(4):217–219, 2022.
- [158] Sam Fankhauser, Stephen M Smith, Myles Allen, Kaya Axelsson, Thomas Hale, Cameron Hepburn, J Michael Kendall, Radhika Khosla, Javier Lezaun, Eli Mitchell-Larson, et al. The meaning of net zero and how to get it right. *Nature Climate Change*, 12(1):15–21, 2022.
- [159] United States Environmental Protection Agency. Understanding global warming potentials, 2022.
- [160] Intergovernmental Panel on Climate Change. Climate change 2021: The physical science basis. contribution of working group I to the sixth assessment report of the intergovernmental panel on climate change. 2021.
- [161] Raymond L Desjardins, Devon E Worth, Xavier PC Vergé, Dominique Maxime, Jim Dyer, and Darrel Cerkowniak. Carbon footprint of beef cattle. *Sustainability*, 4(12):3279–3301, 2012.
- [162] Jasmine A Dillon, Kim R Stackhouse-Lawson, Greg J Thoma, Stacey A Gunter, C Alan Rotz, Ermias Kebreab, David G Riley, Luis O Tedeschi, Juan Villalba, Frank Mitloehner,

- et al. Current state of enteric methane and the carbon footprint of beef and dairy cattle in the United States. *Animal Frontiers*, 11(4):57–68, 2021.
- [163] C Navarrete-Molina, CA Meza-Herrera, MA Herrera-Machuca, N Lopez-Villalobos, A Lopez-Santos, and FG Veliz-Deras. To beef or not to beef: Unveiling the economic environmental impact generated by the intensive beef cattle industry in an arid region. *Journal of Cleaner Production*, 231:1027–1035, 2019.
- [164] Pedro Henrique Presumido, Fernando Sousa, Artur Gonçalves, Tatiane Cristina Dal Bosco, and Manuel Feliciano. Environmental sustainability in beef production and life cycle assessment as a tool for analysis. *U. Porto Journal of Engineering*, 6(1):11–25, 2020.
- [165] Andrea Vitali, Giampiero Grossi, Giuseppe Martino, Umberto Bernabucci, Alessandro Nardone, and Nicola Lacetera. Carbon footprint of organic beef meat from farm to fork: A case study of short supply chain. *Journal of the Science of Food and Agriculture*, 98(14):5518–5524, 2018.
- [166] Lisbeth Mogensen, John E Hermansen, Niels Halberg, Randi Dalgaard, JC Vis, and B Gail Smith. *Life cycle assessment across the food supply chain*, volume 35, pages 115–144. Wiley Online Library, 2009.
- [167] Ecochain. Life cycle assessment (LCA) – everything you need to know. *Ecochain*, pages 1–15, July 2023. Accessed: 2023-07-26.
- [168] X-C Zhang, W-Z Liu, Z Li, and J Chen. Trend and uncertainty analysis of simulated climate change impacts with multiple GCMs and emission scenarios. *Agricultural and Forest Meteorology*, 151(10):1297–1304, 2011.
- [169] Greenhouse gases equivalencies calculator: Calculations and references by the U.S. environmental protection agency, 2021. Available at: <https://www.epa.gov/energy/greenhouse-gases-equivalencies-calculator-calculations-and-references> (Accessed: 2022-06-04).
- [170] Protection of environment, greenhouse gas reporting program: Stationary fuel combustion sources (code of federal regulations), 2016. Available at: <https://www.law.cornell.edu/cfr/text/40/98.33> (Accessed: 2022-09-04).
- [171] Environmental offset solar power, 2021. Available at: <https://freedomforever.com/blog/environmental-offset-solar-power/> (Accessed: 2022-10-04).
- [172] Annual energy outlook 2021 of the us energy information administration, 2021. Available at: https://www.eia.gov/outlooks/aeo/pdf/AEO_Narrative_2021.pdf (Accessed: 2023-07-04).
- [173] Roberto De Vivo and Luigi Zicarelli. Influence of carbon fixation on the mitigation of greenhouse gas emissions from livestock activities in Italy and the achievement of carbon

- neutrality. *Translational Animal Science*, 5(3):txab042, 2021.
- [174] Rakesh Kumar, S Karmakar, Asisan Minz, Jitendra Singh, Abhay Kumar, and Arvind Kumar. Assessment of greenhouse gases emission in maize-wheat cropping system under varied n fertilizer application using cool farm tool. *Frontiers in Environmental Science*, page 355, 2021.
- [175] Felix Adom, Charles Workman, Greg Thoma, and David Shonnard. Carbon footprint analysis of dairy feed from a mill in michigan, USA. *International Dairy Journal*, 31:S21–S28, 2013.
- [176] Carbon Cloud. Product reports, 2023. Available at: <https://apps.carboncloud.com/climatehub/product-reports/id/58994607465> (Accessed: 2024-03-04).
- [177] Mari Rajaniemi, Hannu Mikkola, and Jukka Ahokas. Greenhouse gas emissions from oats, barley, wheat and rye production. *Agronomy Research*, 9:189–195, 2011.
- [178] Amy Quinton. Making cattle more sustainable, 2023. Available at: <https://www.ucdavis.edu/food/news/making-cattle-more-sustainable> (Accessed: 2023-07-26).
- [179] Horacio A Aguirre-Villegas and Rebecca A Larson. Evaluating greenhouse gas emissions from dairy manure management practices using survey data and lifecycle tools. *Journal of Cleaner Production*, 143:169–179, 2017.
- [180] Maria José Cuertos, E Judith Martinez, Rubén Moreno, Rubén Gonzalez, Marta Otero, and Xiomar Gomez. Enhancing anaerobic digestion of poultry blood using activated carbon. *Journal of Advanced Research*, 8(3):297–307, 2017.
- [181] Carbon ecological footprint calculators: Plastic carbon footprint (8 billion trees), 2023. Available at: <https://8billiontrees.com/carbon-offsets-credits/carbon-ecological-footprint-calculators/plastic-carbon-footprint> (Accessed: 2024-01-05).
- [182] Life-cycle carbon footprint analysis of pulp and paper grades in the United States using production line-based data and integration (North Carolina State University), 2023. Available at: <https://shorturl.at/gY5gL> (Accessed: 2024-01-03).
- [183] Carbon footprint of a cardboard box (consumer ecology), 2023. Available at: <https://consumerecology.com/carbon-footprint-of-a-cardboard-box/> (Accessed: 2024-01-04).
- [184] Foodprint Chapter 2: Carbon footprints of foods (University of California, Los Angeles), Oct 2019. Available at: <https://healthy.ucla.edu/wp-content/uploads/2019/10/Foodprint-Chapter-2-Carbon-Footprints-of-Foods-Oct-2019.docx> (Accessed: 2022-02-02).

- [185] Dewayne L Ingram. Life cycle assessment to study the carbon footprint of system components for colorado blue spruce field production and use. *Journal of the American Society for Horticultural Science*, 138(1):3–11, 2013.
- [186] Frank Brentrup, Antoine Hoxha, and Bjarne Christensen. Carbon footprint analysis of mineral fertilizer production in europe and other world regions. In *Conference Paper, The 10th International Conference on Life Cycle Assessment of Food (LCA Food 2016)*, 2016.
- [187] Ramona Cech, Friedrich Leisch, and Johann G Zaller. Pesticide use and associated greenhouse gas emissions in sugar beet, apples, and viticulture in austria from 2000 to 2019. *Agriculture*, 12(6):879, 2022.
- [188] Optimized Thermal Systems, Inc. CA VRF emissions study, 2019. Available at: https://www.optimizedthermalsystems.com/images/pdf/about/CA_VRF_Emissions_Study_rev.pdf (Accessed: 2023-02-02).
- [189] Shariful Kibria Nabil, Sean McCoy, and Md Golam Kibria. Comparative life cycle assessment of electrochemical upgrading of CO₂ to fuels and feedstocks. *Green Chemistry*, 23(2):867–880, 2021.
- [190] Clariant innovates highly effective, low carbon footprint surfactants for personal care and cleaning products, 2023. Available at: <https://shorturl.at/0W9XN> (Accessed: 2024-01-14).
- [191] Carbon footprint of household cleaners, 2017. Available at: <https://theecoguide.org/carbon-footprint-household-cleaners> (Accessed: 2023-02-03).
- [192] Bevan Griffiths-Sattenspiel and Wendy Wilson. The carbon footprint of water. *River Network, Portland*, 2009.
- [193] Angelina Frankowska, Ximena Schmidt Rivera, Sarah Bridle, Alana Marielle Rodrigues Galdino Kluczkowski, Jacqueline Tereza da Silva, Carla Adriano Martins, Fernanda Rauber, Renata Bertazzi Levy, Joanne Cook, and Christian Reynolds. Impacts of home cooking methods and appliances on the GHG emissions of food. *Nature Food*, 1(12):787–791, 2020.
- [194] The CO₂ list of the carbon trust, 2021. Available at: <http://www.co2list.org/files/carbon.htm> (Accessed: 2023-02-03).
- [195] Edgar Blanco and Yossi Sheffi. Carbon emissions in supply chains: environmental, policy and logistics drivers. *Transportation Research Part D: Transport and Environment*, 39:14–30, 2016.
- [196] Ayesha Tandon. ‘Food Miles’ Have Larger Climate Impact Than Thought, Study Suggests. *Carbon Brief*, June 2022. Accessed: 2023-07-26.

- [197] Food and Agriculture Organization. New FAO analysis reveals carbon footprint of agri-food supply chain. FAO News, 2021. Available at: <https://www.fao.org/family-farming/detail/en/c/1458145/> (Accessed: 2023-03-04).
- [198] USDA. Meat and poultry supply chain, 2021. Available at: <https://www.usda.gov/meat> (Accessed: 2023-10-04).
- [199] Jens Burchardt, Michel Frédeau, Miranda Hadfield, Patrick Herhold, Chrissy O’Brien, Cornelius Pieper, and Daniel Weise. Supply chains as a game-changer in the fight against climate change. *Boston Consulting Group*, 2021.
- [200] Shuya Feng, Meisam Mohammady, Han Wang, Xiaochen Li, Zhan Qin, and Yuan Hong. Dpi: Ensuring strict differential privacy for infinite data streaming. *arXiv preprint arXiv:2312.04738*, 2023.
- [201] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.
- [202] Kai Hu, Yaogen Li, Min Xia, Jiasheng Wu, Meixia Lu, Shuai Zhang, and Liguang Weng. Federated learning: a distributed shared machine learning method. *Complexity*, 2021:1–20, 2021.
- [203] Nikolaos Pitropakis, Emmanouil Panaousis, Thanassis Giannetsos, Eleftherios Anastasiadis, and George Loukas. A taxonomy and survey of attacks against machine learning. *Computer Science Review*, 34:100199, 2019.
- [204] Hiroyuki Ito, Ken-Ichi Takeda, Korkut Kaan Tokgoz, Ludovico Minati, Masamoto Fukawa, Chao Li, Jim Bartels, Ikumi Rachi, and Sihan Ai. Japanese Black Beef Cow Behavior Classification Dataset, Jan 2022. Available at: <https://doi.org/10.5281/zenodo.5849025> (Accessed: 2023-03-10).
- [205] Grant Edwards, Racheal H Bryant, Neil P Smith, Helen Hague, Anita Fleming, and Lydia Jane Farrell. Milk production and urination behaviour of dairy cows grazing diverse and simple pastures. volume 75, pages 79–83. New Zealand Society of Animal Production (Inc), 2015.
- [206] J Dijkstra, O Oenema, JW Van Groenigen, JW Spek, AM Van Vuuren, and A Bannink. Diet effects on urine composition of cattle and N₂O emissions. *Animal*, 7(s2):292–302, 2013.
- [207] Anand Kumar Sahu. Cattle breeds dataset. Version 1.0, Kaggle, 2023. Available at: <https://www.kaggle.com/datasets/anandkumarsahu09/cattle-breeds-dataset> (Accessed: 2023-12-26).

- [208] Oguzhan Ulucan, Diclehan Karakaya, and Mehmet Turkan. Meat quality assessment based on deep learning. In *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–5. IEEE, 2019.
- [209] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, et al. FedML: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.