

1 SEAXCHANGE: A BLOCKCHAIN DRIVEN APP FOR
2 TUNA SUPPLY CHAIN MANAGEMENT

3 A Special Problem
4 Presented to
5 the Faculty of the Division of Physical Sciences and Mathematics
6 College of Arts and Sciences
7 University of the Philippines Visayas
8 Miag-ao, Iloilo

9 In Partial Fulfillment
10 of the Requirements for the Degree of
11 Bachelor of Science in Computer Science by

12 BAT-OG, Jeff Rouzel
13 CAHILIG, Maxinne Gwen
14 GANIT, Zyrex Djewel

15 Francis DIMZON, Ph.D.
16 Adviser

17 May 19, 2025

18

Approval Sheet

19

The Division of Physical Sciences and Mathematics, College of Arts and
Sciences, University of the Philippines Visayas

20

certifies that this is the approved version of the following special problem:

21

THIS IS THE TITLE OF YOUR SPECIAL PROBLEM

22

Approved by:

23

Name

Signature

Date

Francis D. Dimzon, Ph.D.

(Adviser)

Ara Abigail E. Ambita

24

(Panel Member)

Christi Florence C. Cala-or

(Panel Member)

Kent Christian A. Castor

(Division Chair)

25 Division of Physical Sciences and Mathematics
26 College of Arts and Sciences
27 University of the Philippines Visayas

28 **Declaration**

29 We, Jeff Rouzel Bat-og, Maxinne Gwen Cahilig and Zyrex Djewel Ganit,
30 hereby certify that this Special Problem has been written by us and is the record
31 of work carried out by us. Any significant borrowings have been properly acknowl-
32 edged and referred.

Name	Signature	Date
Jeff Rouzel Bat-og (Student)	_____	_____
Maxinne Gwen Cahilig (Student)	_____	_____
Zyrex Djewel Ganit (Student)	_____	_____

Dedication

35 This Special Problem, SeaXChange: A Blockchain-Driven App for Tuna Sup-
36 ply Chain Management, is dedicated to those who were instrumental in its creation
37 and purpose:

38 To our esteemed adviser and mentor, **Francis D. Dimzon, Ph.D.**, for his
39 invaluable guidance, expertise, and unwavering support throughout our research
40 journey.

41 To the members of the **Miagao community**, particularly the fishermen, sup-
42 pliers, and retailers from **Barangay Sapa**, and the consumers from **Barangay**
43 **Mat-y**—your daily experiences and challenges in ensuring transparent pricing,
44 product tracking, and maintaining fish product quality served as the foundational
45 motivation for this study. Your participation and insights were crucial.

46 To **Engr. Noel Lucero**, whose valuable perspectives during interviews pro-
47 vided essential context for understanding current trends and issues in the tuna
48 supply chain and the fishing industry as a whole.

49 To our fellow Special Problem classmates and colleagues—for the collabora-
50 tion, shared challenges, and mutual support that made this academic endeavor
51 more enriching.

52 And most importantly, to our beloved families—for their constant love, encour-
53 agement, understanding, and sacrifices that enabled us to pursue and complete
54 this work.

55 This Special Problem paper stands as a testament to your support and as a

⁵⁶ contribution toward addressing the challenges faced within the tuna supply chain.

57

Acknowledgment

58 Our deepest appreciation goes to our thesis adviser, **Francis D. Dimzon**,
59 **Ph.D.**, for his expertise, patient guidance, and unwavering support. We are also
60 grateful to **Christi Florence C. Cala-or**, panelist, for her insightful feedback
61 and suggestions that enriched this study.

62 We thank the **University of the Philippines Visayas**, particularly the **Di-**
63 **sion of Physical Sciences and Mathematics**, College of Arts and Sciences,
64 for providing the academic foundation and resources. We also acknowledge the
65 **Computer Science Faculty** for equipping us with essential skills in software
66 engineering.

67 Special thanks are extended to the **Institute of Marine Fisheries and**
68 **Oceanology**. We particularly recognize **Ricardo P. Babaran Ph.D.**, , for his
69 valuable insights into the state of the tuna supply chain, existing technologies,
70 and the importance of supporting marginalized groups within it. We also thank
71 **Carmelo Del Castillo, Ph.D.**, for sharing related insights from prior thesis
72 work on the tuna supply chain.

73 To our fellow Special Problem classmates and colleagues—thank you for the
74 shared learning experiences, collaboration, and mutual support that made this
75 academic journey more fulfilling.

76 The successful implementation of SeaXChange was made possible through the
77 use of several tools and platforms. We acknowledge **Google Cloud Platform**
78 for hosting our backend services and blockchain network, and the **Hyperledger**
79 **Fabric** open-source framework, along with its documentation and community, for

80 providing a robust foundation for our blockchain system.

81 We extend our gratitude to the community participants and organizations in
82 **Miagao** who generously shared their time and perspectives. Our sincere thanks
83 go to the respondents—fisherfolk, suppliers, retailers, and consumers—whose real-
84 world experiences were vital to this study. We especially thank **Jerome F. Ca-**
85 **batuan** and **Veronica Jeruta** for facilitating access and participation within
86 their respective communities. We also acknowledge **Jagnee Fishing Corpora-**
87 **tion** and **Engr. Noel Lucero** for sharing their knowledge of the fish industry,
88 fishing vessels, and the tuna supply chain.

89 Finally, and most importantly, we extend our heartfelt thanks to our families
90 for their unwavering love, encouragement, sacrifices, and patience throughout the
91 duration of this study.

92 This thesis is a culmination of the support and collaboration of all those men-
93 tioned. We hope that this work contributes meaningfully to improving trans-
94 parency and efficiency in the tuna supply chain.

Abstract

96 The tuna supply chain faces critical challenges regarding traceability, transparency
97 and sustainability, particularly due to certain issues such as illegal, unreported and
98 unregulated (IUU) fishing. Ensuring the traceability within the tuna supply chain
99 is a critical role in enhancing consumer confidence, transparency and promoting
100 adherence to environmental and legal standards. This research explores the appli-
101 cation of blockchain technology as a solution to these given issues. By combining
102 qualitative insights gathered from different key stakeholders across the supply
103 chain, the researchers evaluated the potential of blockchain to improve product
104 traceability, accountability, and trust. The findings suggests that blockchain offers
105 a secure and transparent method of recording the journey of tuna products from
106 catch to market, helping to combat IUU fishing and promote sustainable practices.
107 However, successful implementation requires overcoming barriers related to tech-
108 nological integration, cost, and stakeholder collaboration. This study provides
109 valuable insights into the feasibility and impact of blockchain adoption within
110 other fish supply chains, contributing to the development of more transparent,
111 responsible, and sustainable tuna industries.

112 Suggested keywords based on ACM Computing Classification system can be found
113 at https://dl.acm.org/ccs/ccs_flat.cfm

114 **Keywords:** Blockchain, Traceability, Smart Contract, Supply Chain

¹¹⁵ **Contents**

¹¹⁶	1 Introduction	¹
¹¹⁷	1.1 Overview	¹
¹¹⁸	1.2 Problem Statement	²
¹¹⁹	1.3 Research Objectives	³
¹²⁰	1.3.1 General Objective	³
¹²¹	1.3.2 Specific Objectives	³
¹²²	1.4 Scope and Limitations of the Research	⁴
¹²³	1.5 Significance of the Research	⁴
¹²⁴	2 Review of Related Literature	⁷
¹²⁵	2.1 State of Tuna Industry in the Philippines	⁸
¹²⁶	2.2 Fishing Regulations in the Philippines	⁹

127	2.3 Tuna and Fish Supply Chain	11
128	2.4 Tuna Supply Chain Stages and Roles	12
129	2.5 Factors Affecting the Tuna Supply Chain	14
130	2.6 Technology of Blockchain	16
131	2.7 Opportunities of Blockchain Technology for Supply Chain Management	17
132		
133	2.8 Supply Chain Model with Blockchain Technology of Fishing Industry in Indonesia	20
134		
135	2.9 Existing Technology Intended for Traceability and Supply Chain .	21
136	2.10 Developing a Traceability System for Tuna Supply Chains	24
137	2.11 Chapter Summary	25
138		
139	2.11.1 Comparison Table of Related Studies	26
140		
141	3 Research Methodology	29
142	3.1 Software Development	29
143	3.2 Research Activities	30
144		
	3.2.1 Feedback Collection Method	30

CONTENTS xi

145	3.2.2 Data Gathering	31
146	3.2.3 Designing and Developing the System	32
147	3.2.4 Implementing Algorithms and Services	33
148	3.2.5 Modeling the System Architecture	35
149	4 Results and Discussions	41
150	4.1 Overview	41
151	4.2 Smart Contract Deployment and Installation	42
152	4.2.1 Hyperledger Fabric Prerequisites	42
153	4.2.2 Invoking the Blockchain System	44
154	4.3 Backend Security Analysis (Hyperledger Fabric on GCP)	48
155	4.3.1 System Architecture and Deployment Overview	48
156	4.3.2 Blockchain Network Security	48
157	4.3.3 Smart Contract Automated Test Result	50
158	4.3.4 GCP Infrastructure Security	53
159	4.3.5 Threat Model and Mitigations	55
160	4.4 Mockups	55
161	4.5 Operational Flow of the Web Application	58

162	4.5.1 Landing Page	58
163	4.5.2 Sign Up Page	61
164	4.5.3 Login Page	61
165	4.5.4 Homepage	62
166	4.5.5 Profile	64
167	4.5.6 Logout	64
168	4.6 System Discussion	65
169	4.7 User Demonstration and Feedback Results	66
170	4.7.1 Demo Setup and Scenario	66
171	4.7.2 Summarized Feedback	71
172	4.7.3 Results and Analysis	76
173	5 Conclusion	79
174	5.1 Overview	79
175	5.2 Conclusion	79
176	5.3 Recommendations	80
177	5.4 Summary	81
178	6 References	83

CONTENTS

xiii

179	A Code Snippets	91
180	A.1 Hyperledger Fabric Deployment Instructions	103
181	A.1.1 Environment Setup	103
182	A.1.2 Network Startup and Smart Contract Deployment	104
183	A.1.3 Testing the Smart Contract	106
184	A.1.4 Important Notes	107
185	B Resource Persons	109
186	C Interview Request Letter	111
187	D App Demo Documentation	113

¹⁸⁸ **List of Figures**

¹⁸⁹	3.1 Blockchain Architecture of SeaXChange	36
¹⁹⁰	3.2 Overall System Architecture of SeaXChange	37
¹⁹¹	3.3 Use case diagram for SeaXChange.	39
¹⁹²	4.1 Query Smart Contract: Check assets	45
¹⁹³	4.2 Invoke Smart Contract: Create/Add new tuna asset	45
¹⁹⁴	4.3 Query Smart Contract: Check assets with new tuna asset	45
¹⁹⁵	4.4 Invoke Smart Contract: Transfer asset to a supplier	46
¹⁹⁶	4.5 Query Smart Contract: Check asset after transfer	46
¹⁹⁷	4.6 Invoke Smart Contract: Transfer asset to a retailer	46
¹⁹⁸	4.7 Query Smart Contract: Check asset after transfer	47
¹⁹⁹	4.8 Query Smart Contract: Check updated assets	47
²⁰⁰	4.9 Initialization Confirmation of the Ledger	50

201	4.10 Initialization Confirmation of the Ledger	51
202	4.11 SeaXChange Mockups showing the Authentication Page, Role-Based	
203	Homepage, Asset Transfer Interfaces for Fishermen, Suppliers, and	
204	Retailers, and Asset Viewing page for the Consumers	58
205	4.12 Landing Page	59
206	4.13 Mobile View: Landing Page	60
207	4.14 SignUp Page	61
208	4.15 LogIn Page	61
209	4.16 Fisher Homepage	62
210	4.17 Supplier Homepage	63
211	4.18 Retailer Homepage	63
212	4.19 Consumer Homepage	64
213	4.20 View Profile	64
214	4.21 Log Out	65
215	4.22 Add Catch (Asset)	67
216	4.23 Save Details	68
217	4.24 After Save Details	69
218	4.25 Send Asset	70

LIST OF FIGURES

xvii

219	C.1 Scanned Interview Request Letter	112
220	D.1 Respondents (Fishermen, supplier, and retailer) from Barangay	
221	Sapa, Miagao	113
222	D.2 Respondents (Retailer, and consumer) from Barangay Mat-Y, Miagao	114

List of Tables

224	2.1 Comparison of Studies on Technology for Traceability and Supply	
225	Chain Management	26
226	4.1 Potential Threat and Mitigation	55
227	4.2 Mean ratings and descriptions for functionality-related features per	
228	stakeholder group.	71
229	4.3 Mean ratings and descriptions for end-user needs-related features	
230	per stakeholder group.	72
231	4.4 Mean ratings and descriptions for performance-related features per	
232	stakeholder group.	72
233	4.5 Mean ratings and descriptions for usability-related features per	
234	stakeholder group.	73
235	4.6 Mean ratings and descriptions for ease of use-related features per	
236	stakeholder group.	75

237	4.7 Mean ratings and descriptions for feasibility-related features per	
238	stakeholder group.	75

²³⁹ Chapter 1

²⁴⁰ Introduction

²⁴¹ 1.1 Overview

²⁴² The tuna supply chain faces critical issues that affect both the industry and its
²⁴³ consumers. Illegal fishing, overfishing, and poor traceability threaten the sustain-
²⁴⁴ ability and ethical trade. A lack of transparency compromises product quality
²⁴⁵ and misleads consumers. Blockchain technology can address these challenges by
²⁴⁶ tracking the tuna's journey from ocean to plate through a secure, tamper-proof
²⁴⁷ ledger. This system also holds stakeholders to comply to legal standards.

²⁴⁸ Although blockchain integration in the tuna sector is underway, gaps remain in its
²⁴⁹ implementation. This study helps to address the industry's need for transparent
²⁵⁰ and secure tracking of tuna products from ocean to consumer, while assessing the
²⁵¹ feasibility of implementing blockchain at scale in the seafood sector.

²⁵² 1.2 Problem Statement

²⁵³ Fish is one of the most consumed protein products in the Philippines. Before the
²⁵⁴ COVID-19 pandemic, fish was the most consumed animal protein in the Philip-
²⁵⁵ pines with annual per capita consumption of 40 kg (Agriculture and Agri-Food
²⁵⁶ Canada, 2022). Among the wide variety of fish, tuna stands out as a particularly
²⁵⁷ significant commodity due to its high demand and economic value. Although the
²⁵⁸ nation is gifted with an abundance of aquatic resources, the methods of dispersal
²⁵⁹ of the product usually leads to inefficiency in terms of sales, pricing, and overall
²⁶⁰ product quality for the consumers. The problem arises with the introduction of
²⁶¹ a supply chain from several middlemen between the fish farmer or producer and
²⁶² wholesaler in the coastal and aquatic regions towards the consumers with less ac-
²⁶³ cess to fresh fish. Consequently, both consumers and suppliers struggle to ensure
²⁶⁴ transparent and fair pricing, accurately track tuna products, and maintain consis-
²⁶⁵ tent quality. These challenges stem from inefficient methods of product dispersal
²⁶⁶ and the presence of multiple intermediaries between fisherfolks in coastal munici-
²⁶⁷ palities and consumers. This fragmented supply chain introduces delays, obscures
²⁶⁸ product origin, and opens opportunities for mishandling or mislabeling. To ad-
²⁶⁹ dress this, the study focuses on the absence of a reliable, real-time traceability
²⁷⁰ system that allows stakeholders to verify the origin, novelty, handling processes,
²⁷¹ and adherence to ethical sourcing practices of tuna at every stage. Without such
²⁷² a system, the tuna supply chain and industry lack the transparency and account-
²⁷³ ability necessary to build consumer trust and improve supply chain efficiency. By
²⁷⁴ introducing a blockchain-driven solution, this study aims to streamline the trace-
²⁷⁵ ability process and reduce the negative impact of intermediary-heavy distribution.

²⁷⁶ **1.3 Research Objectives**

²⁷⁷ **1.3.1 General Objective**

²⁷⁸ The general objective of the study is to design and develop a blockchain-driven
²⁷⁹ application that improves the traceability of the tuna supply chain. Given the
²⁸⁰ timely issues regarding illegal, unreported and unregulated (IUU) fishing and lack
²⁸¹ of product traceability and transparency, this study addresses these challenges
²⁸² through an application that has the capability to provide an immutable ledger
²⁸³ and tamper-proof records. The result of this study serves as a framework for
²⁸⁴ integrating blockchain technology in the fish supply chain, specifically for tuna.
²⁸⁵ This also supports future researchers and developers facing similar challenges in
²⁸⁶ the industry.

²⁸⁷ **1.3.2 Specific Objectives**

²⁸⁸ To further specify the research objectives, the study focuses to:

- ²⁸⁹ 1. develop a smart contract framework using blockchain technology for data
²⁹⁰ verification and transaction recording, ensuring secure and tamper-proof
²⁹¹ data for the stakeholders,
- ²⁹² 2. design and develop a blockchain-driven application with a user-friendly in-
²⁹³ terface that allows stakeholders to access and input data while enhancing
²⁹⁴ traceability in the tuna supply chain through a tuna asset record for the
²⁹⁵ supply chain participants, and

296 3. deploy the application after completing all necessary preparations for soft-
297 ware development and to evaluate its overall results and effectiveness in en-
298 hancing the tuna supply chain, as discussed in the fourth and fifth chapters
299 of the study.

300 1.4 Scope and Limitations of the Research

301 The scope of this study focuses on how blockchain technology can be applied to
302 enhance traceability and transparency within the tuna supply chain. It involves
303 features such as smart contracts for recording the transactions and user interface
304 for stakeholders. The study also focuses exclusively on whole, small-sized caught
305 tuna products in the supply chain, excluding processed forms such as canned
306 or packaged tuna. The research examines the traceability of whole tuna from
307 capture to market sale, specifically centering on a supplier based in Miagao and
308 San Joaquin, Iloilo.

309 This study focuses solely on the supply chain within the specified municipalities
310 of Iloilo, so findings may not fully represent global practices. As it concentrates
311 on blockchain's role in traceability, other potential functions of the technology fall
312 outside of the scope of this research.

313 1.5 Significance of the Research

314 This study serves a significant purpose for several stakeholders in the tuna supply
315 chain. This study aims to solve the problems related to the management of tuna

³¹⁶ supply chain, particularly with regards to product traceability.

³¹⁷ • The Stakeholders

³¹⁸ – This study enhances transparency and accountability, allowing stakeholders such as fishers, suppliers, and retailers to access tamper-proof and accurate information, thereby promoting a more ethical and authentic supply chain. By providing a digital record of the product's history, this study helps ensure compliance with environmental and legal standards. In cases of anomalies such as oil spills, red tide occurrences, and illegal fishing activities, stakeholders can be involved in identifying and addressing these issues, fostering a collaborative approach to sustainability. Similarly, the record of a tuna asset can be utilized for accountability purposes when problems such as damaged products, mislabeling, or contamination arise, allowing stakeholders to trace and resolve them efficiently.

³³⁰ • The Consumers

³³¹ – This study helps consumers verify the history of the tuna product from its origin up until its journey to the consumers, therefore increasing trust and transparency. By promoting traceability, it offers a more detailed and verifiable record of the supply chain, enabling consumers to assess sustainability practices and identify stakeholders responsible for any potential issues affecting the tuna product. This, in turn, encourages critical evaluation of the tuna supply chain, driving improvements in accountability, resource management, and ethical sourcing.

³³⁹ • For Future Researchers

³⁴⁰ – As blockchain technology continues to grow, this study contributes to
³⁴¹ the application of blockchain in the supply chain management and the
³⁴² insights regarding its benefits and limitations. This research can be
³⁴³ helpful in the growing knowledge on digital solutions for traceability
³⁴⁴ and transparency for future research.

³⁴⁵ • The Policy Makers

³⁴⁶ – This study provides policy makers a reliable and data-driven founda-
³⁴⁷ tion in monitoring and regulating the tuna supply chain. By lever-
³⁴⁸ aging tamper-proof and transparent records, policymakers can more
³⁴⁹ effectively enforce compliance with fishing quotas and environmental
³⁵⁰ protections. This research also aids in lessening the illegal, unreported
³⁵¹ and unregulated (IUU) fishing practices, contributing to the national
³⁵² sustainability goals.

³⁵³ Chapter 2

³⁵⁴ Review of Related Literature

³⁵⁵ This chapter reviews existing literature related to the traceability of the tuna sup-
³⁵⁶ ply chain and the potential application of blockchain technology. In purchasing
³⁵⁷ goods, one important consideration is the product quality. An important factor
³⁵⁸ of determining the quality is to know the traceability of the supply chain. Trace-
³⁵⁹ ability refers to the ability of tracking the journey of the product from its source
³⁶⁰ until its destination. The term “traceability” is now more utilized in both the
³⁶¹ food and production industry (Islam & Cullen, 2021). In the context of the tuna
³⁶² supply chain, it can be used not only to promote transparency to consumers but to
³⁶³ also ensure compliance with environmental and legal standards. With blockchain
³⁶⁴ technology, the status of tuna at each stage could be recorded in the blockchain
³⁶⁵ which could be used for traceability. This paper aims to address the following
³⁶⁶ research question: How can blockchain technology improve the traceability of the
³⁶⁷ tuna supply chain management? To explore this, the chapter reviews literature
³⁶⁸ on the state of the tuna industry in the Philippines, fishing regulations, and the

369 structure of the tuna supply chain, including its stages and the roles of key actors.
370 It also examines factors that affect the efficiency and transparency of the supply
371 chain. The discussion then turns to blockchain technology and its potential ap-
372 plications in supply chain management, with examples from the Philippines and
373 Indonesia. Finally, the chapter reviews existing traceability technologies and sys-
374 tems and concludes with key insights that inform the development of this study's
375 proposed solution.

376 In purchasing goods, one thing to consider is the quality of it. An important part
377 of determining the quality is to know the traceability of the supply chain. Trace-
378 ability refers to the ability of tracking the journey of the product from its source
379 until its destination. The term “traceability” is now more utilized in both the
380 food and production industry (Islam & Cullen, 2021). In the context of the tuna
381 supply chain, it can be used not only to promote transparency to consumers but to
382 also ensure compliance with environmental and legal standards. With blockchain
383 technology, the status of tuna at each stage could be recorded in the blockchain
384 which could be used for traceability. This paper aims to address the following
385 research question: *How can blockchain technology improve the traceability of the*
386 *tuna supply chain management?*

387 2.1 State of Tuna Industry in the Philippines

388 In 2014, the Philippines became the top global producer of tuna according to
389 Llanto, et. al. (2018). The tuna is caught in domestic and international fishing
390 grounds near the country through various fishing methods such as purse seines,

391 gill nets, handline (hook and line) and ring net. Among the tuna species, the
392 skipjack tuna accounted for the largest portion of the catch by 40%. The study of
393 PCMARD (1993 as cited in Nepomuceno et al., 2020) stated that skipjack tuna
394 are often caught out in open waters or in offshore areas. In addition, Nepomuceno
395 et al. (2020) mentioned in their study that the dominant production of skipjack
396 tuna, together with yellowfin tuna, was recorded in South Cotabato. The tuna
397 supply has declined since 2000 due to various factors including overfishing, climate
398 change, and the laws and regulations imposed by different governing bodies for
399 the tuna fishing ground such as the Regulation No. 56, released by the Indone-
400 sian Maritime Affairs and Fisheries Ministry in November 2014. The regulation
401 imposed a moratorium on issuance of fishing licenses from 3 November 2014 to 30
402 April 2015 to eliminate illegal, unreported, and unregulated fishing in Indonesian
403 waters near Mindanao where tuna are known to thrive (Llanto et al, 2018). The
404 regulation imposed for the protection of tuna fishing grounds in the western and
405 central pacific ocean also lead to the decline of local tuna production which re-
406 quires the fishing operators of the Philippines to invest in the manufacturing and
407 processing of fish particularly tuna in Indonesia which includes hiring Indonesian
408 crew to be deployed in the Philippine fishing vessels (Llanto et al, 2018).

409 2.2 Fishing Regulations in the Philippines

410 A study of Asche et al. (2018) divided the fishing management strategies that
411 include right-based fishery management like territorial use of rights, access rights
412 and harvest rights. It discussed that a rights-based system could support the sus-
413 tainability of global fisheries by taking in account the three pillars of sustainability

414 (economic development, social development, and environmental protection) rather
415 than focusing on their trade-offs. A restriction on the fisherman's behavior by har-
416 vest rights and catch shares could be a profit problem for them in the short-run
417 but in the long-run, this could help both in the fish stock and the fishermen's
418 profit. Lack of restriction could lead to overfishing. Access rights limit the en-
419 try to fishery through permits which can also reduce the effect of high harvest
420 levels. A sustainable fishing management system in the Philippines is important
421 in order to preserve marine resources. To preserve these resources and protect
422 the livelihood of local communities, various fishing management strategies should
423 be implemented. A collaboration between the fishermen, local government and
424 other stakeholders often happens to manage marine resources (Pomeroy & Court-
425 ney, 2018). The study of Pomeroy and Courtney discussed that marine tenure
426 refers to the rights and responsibilities in terms of who can access the marine and
427 coastal resources. The 1998 Fisheries Code paved the way for local government
428 units (LGUs) to be involved in the management of municipal waters. LGUs are
429 given the responsibility to overlook and regulate fisheries and establish marine
430 tenure rights for fishers within 15 km from shore and these rights are applicable
431 for municipal fishers and their respective organizations that are listed in the reg-
432 istry (Pomeroy & Courtney, 2018). In this way, it resolved problems in terms of
433 fishing rights between small-scale and commercial fishing.

434 According to the study conducted by Mullon et al. (2017), the five major species of
435 tuna: yellowfin *Thunnus albacares*, bigeye *Thunnus obesus*, bluefin *Thunnus thyn-
436 nus* or *Thunus orientalis*, albacore *Thunnus alalunga*, and skipjack *Katsuwonus
437 pelamis* are harvested to meet the global supply chain demand which causes those
438 group of tuna fishes to be heavily exploited and threatened. The study conducted

439 by Paillin et al. (2022) states that there are multiple risk agents in the supply
440 chain assessment of tuna, these include the lack of standard environmental man-
441 agement system, lack of maintenance management, and lack of quality control
442 from suppliers. The usage of efficient boats and good quality catching technology
443 can also lead to fisheries depletion which causes various agency such as BFAR
444 (Bureau of Fisheries and Aquatic Resources), the local government units, and the
445 Philippine Coast Guard to enable policies for upholding closed fishing season to
446 restrict large scale fishing vessel to minimize the fishing activities in the identified
447 areas (Macusi et al, 2023). The implementation of closed fishing season caused de-
448 lay or lack of fish supply, which led to higher fish prices. The growing demands and
449 depleting population of tuna fishes coupled with the rapid increase in fuel costs
450 can have a negative impact on the future of the supply chain in tuna fisheries
451 (Mullon et al., 2017). With factors concerning the slow decline of tuna catches in
452 the Philippines and surrounding nations, the future of the global supply chain of
453 tuna must be addressed.

454 2.3 Tuna and Fish Supply Chain

455 According to Macusi et al (2023), the implementation of traceability programs in
456 the agricultural product commodities and value chain in the Philippines is slower
457 than its competing nation for tuna production. The Philippines has been steadily
458 responding to the market innovation and integration of cost-effective and smart
459 technologies for the traceability of various commodities. Accurate catch data is
460 crucial for determining the attributes of the fish health, size, volumes, and matu-
461 rity (Grantham et al, 2022) which can be used as a basis for the transparency of

462 the traceability of the fish product. Illegal, unreported, and unregulated (IUU)
463 is another concern for the fish industry. In the 2000s, the persistent IUU became
464 a global crisis affecting the biological, ecological, and socio-economics factors re-
465 volving around marine livelihood in Southeast Asia (Malinee et al, 2020). IUU
466 fishing is known to cause short- and long-term problems in the socio-economic
467 opportunities which affects food security and results in the possible collapse of
468 the fish industry and stocks due to overfishing (Malinee et al, 2020).

469 The establishment of marine protected areas in the Davao Gulf (MPAs) affected
470 the management of small-scale fisheries due to the growing population and de-
471 mands for seafood products. The closure of a wide range of fishing areas hosting
472 diverse and marine organisms has affected the socio-economics and livelihood of
473 the local and small-scale fishermen (Macusi et al, 2023), this in turn resulted
474 in IUU fishing. To ensure that fish stocks in the gulf are sustainably managed,
475 the implementation of GPS for tracking the movement and activities of fishers
476 through logbook and habitat monitoring can provide data and insights for track-
477 ing, monitoring, and understanding the condition of the marine resources (Obura
478 et al, 2019; Macusi et al, 2023).

479 2.4 Tuna Supply Chain Stages and Roles

480 The study conducted by Delfino (2023) highlights the roles of different actors in-
481 volved in the supply, production, distribution, and marketing of skipjack tuna in
482 Lagonoy Gulf in the Philippines. The study showcased a total of eleven intercon-
483 nected value chains but are generalized into four major stages or roles - fishers,

484 wholesalers, retailers, and processors. The fishers are the initial players responsible
485 for catching fish using boats or fishing vessels equipped with purse seines, gillnets,
486 and handlines(hook and line). Wholesalers are the actors for selling freshly caught
487 fish locally and regionally, they receive the fish supply directly from the fishers.
488 The next stage after wholesalers are the retailers, these intermediaries sell the fish
489 product to local markets, house-to-house (*libod*" in Visayan languages), and other
490 local medium such as *talipapa* or fish stands. Another intermediary is the proces-
491 sors, they convert fresh skipjack tuna into products like smoked tuna. The given
492 stages also overlapped in some cases as there are fisher-wholesalers who catch and
493 sell the fishes directly to retailers and there are also retailer-processors that both
494 sell whole and processed products. Despite having a firm system to transport fish
495 from sea to table, all the actors face problems during seasonal challenges involving
496 the availability of the tuna product. The fishers also need to consider strict local
497 regulations such as RA 10654 and RA 8550. The strict implementation of RA
498 10654 and RA 8850 at the local level or the Fisheries Code of the Local Philip-
499 pines aims to curb the problem encountered during season of deficit tuna supply
500 by limiting fishing activities and implementation of 15-km boundary lines in the
501 municipal waters of each municipality (Delfino, 2023). The study suggests that
502 improving conditions for value chain actors, particularly through support services
503 and government involvement could lead to a stable and sustainable exchange of
504 skipjack tuna and other seafood products from sea to table.

505 A study of Digal et al. (2017) discussed one of the value chains which was the
506 purse seine or the skipjack tuna value chain in the Philippines. Purse seining
507 is the method of catching a school of fish wherein it uses a large net around it,
508 trapping them and pulling the bottom of the net like a purse-like (Digal et al.,

509 2017) . This type of catching often targets dense fishes like tuna. Skipjack fishes
510 that weigh 300 grams and above are often sold to canneries, while the smaller ones
511 are sold at local markets, often used for consumption by Filipinos. Purse seiners
512 are usually employees of a fishing company and they have a fixed salary. They
513 could reach international waters so they need to bring their passports with them.
514 Jamboleros, who act as distributors, often buy from different fishing companies
515 per *banyera* or tub. They will then pack the fish and sell it to traders/truckers
516 who go to General Santos fishport. These traders will deliver it to the retailers
517 across Kidapawan who contacted them. There is no formal contract between the
518 jambolero and traders/truckers. One of the issues of the retailer is for everyday
519 that a fish is not sold, they would have a ₱10.00 less per kilogram.

520 2.5 Factors Affecting the Tuna Supply Chain

521 The tuna supply chain faced several factors and challenges for the safety and
522 quality of the product (Mercogliano & Santonicola, 2019). Without the proper
523 handling of the tuna after catching it can lead to various food-borne diseases and
524 outbreaks. The most frequent and mitigated food-borne causing compound is
525 Histamine(HIS) which causes Scombrotoxin fish poisoning (SPF) outbreaks re-
526 lated to food allergies when consumed (EFSA, 2017; Peruzy et al., 2017). Tuna
527 species are known for having high presence of amino acid histidine concentrations
528 which are converted to HIS by bacterial enzyme histidine-decarboxylase or HDS
529 (Aponte et al., 2018; Verkhivker & Altman, 2018). To combat the risk of SPF
530 and other food-borne diseases caused by tuna consumption, several safety hazards
531 and protocols were imposed to the tuna supply chain management. The term cold

532 chain refers to the storing of fish in temperatures less than 5°C after it was caught
533 (Yang & Lin, 2017). According to the article published by Mercogliano and San-
534 tonicola (2019), implementing a cold chain from the time the fish is caught until it
535 is consumed is crucial for mitigating the outbreak of HIS poisoning. Additionally,
536 the article also states that using high-quality raw tuna, cold chain maintenance,
537 pre-cooking, and cooking can also reduce HIS development.

538 According to the study conducted by Pacoma and Yap-Dejeto entitled "Health
539 Risk Assessment: Total Mercury in Canned Tuna and in Yellowfin and Frigate
540 Tuna Caught from Leyte Gulf and Philippine Sea", examines mercury contamina-
541 tion in both canned and locally caught tuna in the Philippines. Methylmercury,
542 a potent neurotoxin, presents risks especially to vulnerable groups like pregnant
543 women and children. The study reveals that canned tuna generally has higher
544 mercury levels (0.07 µg/g) than locally caught tuna (0.002–0.024 µg/g). Based
545 on the Food and Agriculture Organization's fish consumption data, the mercury
546 intake from locally caught tuna is within the World Health Organization's safe
547 limits, whereas canned tuna may exceed these limits for some groups. This high-
548 lights the need for monitoring of the mercury levels in the Philippine tuna supply
549 chain, as tuna is a dietary staple and economic asset in the region, to mitigate
550 health risks from chronic exposure.

551 Risk management is important for tuna supply chains to analyze the root of the
552 risk and to assess the probability of such cases through the information taken from
553 the different locations or sorting states where the tuna product is handled before
554 being purchased by the consumer or end-use state (Parenreng et al, 2016) .

555 2.6 Technology of Blockchain

556 According to Zheng et al. (2017), the idea of blockchain was first introduced in
557 2008 and was first implemented in the Bitcoin business which deals with cryp-
558 tocurrencies. This kind of technology tracks transactions and stores it in a list of
559 blocks. According to Sarmah (2018) , it acts as a database of transactions which
560 is overseen and verified by distributed nodes. Blockchain works by linking blocks
561 (where data is stored). When a transaction is initiated, it is then broadcasted to
562 a network of computers that verifies the transaction and if the verification is suc-
563 cessful, it will then be grouped and linked with the previous transactions that will
564 be added to the blockchain (Zheng et al., 2017) . It does not require a middleman
565 because it operates on a peer-to-peer transaction. This eliminates the traditional
566 way of a central authority like the bank (Sarmah, 2018) .

567 Automated process of transactions is also one of the salient features of blockchain
568 which is executed automatically based on predefined rules involving no third par-
569 ties. These predefined rules are conditions that need to be met in order for the
570 transaction to proceed. Given this, blockchain is described to be a “trust-free
571 technology” since it reduces the number of trusted individuals instead they trust
572 the machine itself which is difficult to manipulate given its cryptographic security
573 (Ali et al., 2023). Blockchain also ensures immutability with its data. Once the
574 data has been added to the blockchain, it is difficult to change since each block has
575 a cryptographic link to other blocks, which guarantees that the data is tamper-
576 proof and permanent (Ali et al., 2023). This also brings back to the essence
577 of blockchain being trustworthy. Immutability and tamper-proof enhances data
578 management. It also helps prevent fraudulent activities especially with finances.

2.7. OPPORTUNITIES OF BLOCKCHAIN TECHNOLOGY FOR SUPPLY CHAIN MANAGEMENT

579 Transparency with transactions is one of the features of blockchain which makes
580 the chances of data being tampered less because it is accessible to anyone on the
581 network (Ali et al., 2023).

582 According to Nasurudeen Ahamed et al. (2020), Blockchain technologies are clas-
583 sified into 3 types: Public Blockchain, Private Blockchain,. Consortium Blockchain.
584 In Public Blockchain, all public peers can join together and have equal rights (for
585 example, read, write, and execute) on the public node. In Private Blockchain,
586 only Authorized Private Peers have access to the network. The access to the node
587 in the private peer is limited to the specific node. In Consortium Blockchain,
588 only the authorized team can access and join this blockchain, and all operations
589 in the node must adhere to the access. Their paper, Sea Food Supply Chain
590 Management Using Blockchain, gave the idea that with the purpose of creating a
591 blockchain-driven application, a public blockchain approach could be appropriate
592 for handling consumer-based information as users can verify non-sensitive data
593 like prices, freshness and availability. While handling sensitive information such
594 as internal works and logistics, a consortium blockchain where authorized users
595 such as fish owners, distributors, manufacturers, etc. can handle the core supply
596 chain operations, like tracking the movement of tuna from catch to market.

597 **2.7 Opportunities of Blockchain Technology for 598 Supply Chain Management**

599 Supply chain is the term used for understanding the business activities for design-
600 ing, developing, delivering, purchasing, and using a product or service (Hugos,

601 2024). Companies and various industries are heavily relying on supply chains to
602 achieve their business objectives. The purpose of supply chain began to be more
603 significant in the last century as firms discovered that supply chain can be used for
604 competitive advantage instead of just a cost driver as believed in the bygone days
605 (Snyder & Shen, 2019). Following the supply chain paradigm can demonstrate the
606 delivery of a product or service while strongly emphasizing the customer's speci-
607 fications. With the increasing studies conducted and published for supply chain,
608 many companies adopted this practice for the benefit of their longevity, as such
609 the term supply chain management has come into place. The Council of Supply
610 Management Professionals or CSCMP (2024) defines supply chain management
611 as the planning and management of all activities involved in sourcing and pro-
612 curement, conversion, and all logistics management activities; essentially, supply
613 chain management integrates supply and demand management within and across
614 the company. Supply chain management is also involved with the relationship
615 with collaborators and channel partners such as suppliers, intermediaries, third
616 party providers, and customers (CSCMP, 2024) .

617 In the article of Cordova et. al (Cordova et. al, 2021), the role of supply chain man-
618 agement and the growing opportunities for blockchain technology in supply chain
619 management was discussed. According to Cordova (2021), the recent innovation
620 and globalization has given rise to the idea of using a data innovation framework
621 for supply chain management. Technologies such as blockchains and enterprise
622 resource planning (ERP) are among the highly contested platforms for supply
623 chain management to operate in a seamless interaction and distribution with the
624 product while heavily relying on modern technology and innovations. The logistic
625 business of the supply chain market is wide and complex, the distribution and

2.7. OPPORTUNITIES OF BLOCKCHAIN TECHNOLOGY FOR SUPPLY CHAIN MANAGEMENT

flow of products is not a simple job, and it heavily relies on paperwork (Georgiou, 2019; Cordova et. al, 2021). The usage of paperwork for logistic business can be at higher risk for lack of transparency, complex or unreliable tracking, deficiency of information, and possible dispute due to the tendency of paper to disappear or tear down, this can in turn delay the process and delivery of the item/product. With the issues encountered in the supply chain market, businesses, people and enterprises are eyeing toward the application of blockchain technology on supply chain management (Cordova et. al, 2021).

Implementing blockchain innovation in ERP systems and companies that use digital platforms can provide opportunities and contribute greatly for business processes (EOS Costa Rica, 2019 as cited by Cordova et al, 2021). The ability of blockchain technology to append new transactions to an existing block containing data can be thought of as a decentralized ledger (Cole et al, 2019). The method of blockchain to behave like a decentralized ledger can serve as a single unified source of data which in turns create a clear and consistent audit trail involved in the manufacturing, assembly, supply, and maintenance processes. According to Cole et al (2019), blockchains provide data to the movement and relation of products from its origin, inventory, shipment, and purchase. One potential of blockchain for supply chain management (Hackius & Petersen, 2017) is the ease of paperwork processing, specifically in ocean freight. When IBM and Maersk settled for a permissioned blockchain solution, they were able to connect a global network of shippers, carriers, ports, and customs. Another potential of blockchain in SCM is to identify counterfeit products. In the pharmaceutical industry and healthcare setting, blockchain could improve patient safety and hazard through establishing supply chain transparency from manufacturers through wholesale and pharma-

651 cies to the individual patients (Hackius & Petersen, 2017). Using blockchain can
652 make it harder to tamper or alter the products chain of origin with illegal and
653 counterfeit products. Blockchain have the potential to facilitate origin tracking.
654 According to Cordova et al (2021) , blockchain allows organizations to input rel-
655 evant data inside a chain which would have constant updates and tracking, this
656 supports visibility and traceability of the origin of the product. Smart contracts,
657 an executable code and a feature of blockchain, serves as a computer protocol
658 made between participants to digitally facilitate, execute, verify, and enforce an
659 agreement or terms of contract which is then stored in the blockchain (Khan et
660 al, 2021).

661 **2.8 Supply Chain Model with Blockchain Tech- 662 nology of Fishing Industry in Indonesia**

663 Larissa and Parung (2021) who explored the application of blockchain and de-
664 signed a supply chain model based on it, specifically for the Indonesian fishing
665 industry, aimed to mitigate the challenges in the fishery industry such as product
666 quality (perishability), long shipping times ,and data manipulation. The model
667 they developed of using QR codes for each player in the supply chain then tracking
668 it by scanning the QR code, could inspire our approach in building a blockchain-
669 driven application for the tuna supply chain in the Philippines.

670 **2.9 Existing Technology Intended for Traceability and Supply Chain**

672 A study of Shamsuzzoha et al. (2023) discussed the feasibility of implementing
673 a blockchain driven application called ‘Tracey’ for monitoring the fish traceability
674 in supply chain management. The study utilized the theoretical framework
675 developed by Islam & Cullen (2021) for improving the understanding and effec-
676 tiveness of implementing a food traceability system. The framework consists of
677 four principles as a basis for the supply chain management: identification, data
678 recording, data integration, and accessibility (Islam & Cullen, 2021). The Tracey
679 application utilized a public-private hybrid blockchain-based conceptual frame-
680 work by Mantravadi and Srai (2023) to uphold the transparency, traceability, and
681 certification of the sea food produce, specifically shrimp. The prototype being
682 studied by Shamsuzzoha et al. (2023) called Tracey focuses on the mobile-based
683 solution approach, the study found that the most widely used smartphone type
684 in the Philippines is the android phone which is where the Tracey prototype is
685 intended to be used. The Tracey app allows fishermen to log their catch details
686 and buyers to verify and update transaction history (Shamsuzzoha et al., 2023).
687 The Tracey app uses a central database for storing fish trading data and a decen-
688 tralized ledger or blockchain for traceability purposes. The decentralized ledger
689 acts as a tamper-proof copy of the data recorded by fishermen and buyers. The
690 result of the study finds that fishermen are open to using digital methods for
691 payments and confidentiality which is required for exporting the fish product to
692 maintain high standards for traceability, catch certification, and product quality.
693 The usage of blockchain as exemplified by the Tracey project can be used for

694 upholding the restriction for IUU due to its ability to ensure transparent trade,
695 consistent records, and accessibility. The result and discussion of the study of
696 Shamsuzzoha et al. (2023) provides a solution for improving the sustainability of
697 tuna fishery and ensures that Filipino fishermen receive fair compensation. For
698 the study limitation of the Tracey project, although there is a high acceptabil-
699 ity of potentially using the app for fishermen, there are still constraints in terms
700 of proper incentives, connectivity issues, technology usability, and education for
701 using the app.

702 A study of Cocco and Mannaro (2021) proposed a blockchain-based technology
703 in the traceability of the supply chain management of a traditional Italian food
704 product, Carasau bread, which is made from durum wheat flour, salt, yeast and
705 water. Since the production of this product is traditional, consumers would de-
706 mand for transparency on the methods used in the Carasau bread production to
707 ensure authenticity. The proposed model involves the combination of the appli-
708 cation of Internet of Things (IoT), specifically the Radio Frequency Identification
709 (RFID) sensors and Interplanetary File System (IPFS) with Blockchain (Cocco
710 & Mannaro, 2021). RFID is a technology that uses radio frequencies to identify
711 and track a tagged object while IPFS allows files to be stored and tracked over a
712 decentralized and distributed file system. Cocco and Mannaro (2021) also stated
713 that using RFID tags that will be integrated with different sensors to monitor
714 food quality will be useful in their study. These sensors include freshness indica-
715 tors to monitor the food quality when packaged, biosensors to detect degradation
716 molecules, time temperature indicator to measure and record temperature and
717 humidity sensors to detect the amount of water vapor in the atmosphere. The
718 integration of IoT and blockchain technology can have a huge impact in increasing

719 traceability in agri-food supply chain. Moreover, this study proposed to have a
720 generic agri-food traceability system which will be based on Ethereum blockchain,
721 Radio-frequency identification (RFID), Near Field Communication (NFC), and
722 Interplanetary File System (IPFS) technology. Moreover, the model proposal also
723 includes sensor network devices, smart contacts, optical cameras and an external
724 database. Each IoT device will be connected to Raspberry Pi and interfaces with
725 blockchain implementing smart contracts and IPFS which authorities can inspect
726 every node and batch online through the uploaded files in IFPS; hashes of the
727 uploaded files on IFPS are also stored on blockchain (Cocco & Mannaro, 2021) .
728 In this way, users along the chain supply can view and trace each batch using the
729 NFC tags promoting transparency and traceability. Overall, the proposed model
730 is a combination of two subsystems. The first one is an on-chain system which is
731 the blockchain implementing smart contracts and will be developed using Solidity,
732 an object-oriented language. The second one is an off-chain system that will be
733 implemented in Javascript using Node.js (to interact with the smart contracts)
734 and Web3.js packages (to interact with blockchain) and these packages should be
735 installed on the Raspberry Pi. However, the purpose of this study is to exam-
736 ine the traceability systems of the agri-food industry and further provide possible
737 solutions.

738 **2.10 Developing a Traceability System for Tuna 739 Supply Chains**

740 The study of Kresna et al. (2017), proposed an IT-based traceability system for
741 tuna supply chain as opposed to the traditional paper based traceability system
742 which has several limitations such as the potency to be manipulated, error by
743 the human, language barrier, and physical damage. The architecture comprises
744 several layers: infrastructure, data, application, communication, and user lay-
745 ers. The infrastructure layer includes computer hardware, network infrastructure,
746 and sensing devices like CCTV, GPS, and RFID for data acquisition. The data
747 layer serves as the system's database, featuring both a main system database and
748 an emergency database for critical situations. The application layer consists of
749 various modules—admin, tracing, transporter, supplier, and government—that
750 registered actors can access through different interfaces. Finally, the user layer
751 consists of the registered actors who utilize the system.

752 The journal article of Tiwari (2020) called Application of Blockchain in Agri-
753 Food Supply Chain conducted two case studies for a blockchain driven app built
754 for supply chain related to food, fishing, and agriculture. The first case-study is
755 the usage and effectiveness of the Provenance system for tuna tracking certifica-
756 tion. The objective of the Provenance system is to enhance transparency in the
757 tuna supply chain by ensuring certification and standard compliance across all
758 roles(e.g. supplier, retailer) in the chain. The system is built using six modular
759 programs: registering, standards, production, manufacturing, tagging, and user-
760 interface. The usage of blockchain in the Provenance system allows transactions to
761 be recorded to allow shared ledger for transparency and smart contracts for secure

762 exchanges of money or information. The usage of the Provenance system is to
763 solve the issues encountered in the tuna fishing industry affected by various factors
764 such as illegal, unregulated, unauthorized (IUU) fishing, fraud, and human rights
765 abuses. The solution of the Provenance system is to allow tracking, tracing, and
766 certification of tuna using blockchain. The Provenance system has a smart tag-
767 ging feature that allows fishermen to use SMS for digital assets on the blockchain
768 to track where the fish, in return, all supply chain stakeholders can access the
769 data that was sourced from the SMS. The second case-study is the usage of the
770 IBM Food Trust for transparency in the food supply chain. The IBM Food Trust
771 aims to solve the problems in the food supply chain, specifically in product safety.
772 Locating supply chain items in real-time using identifiers like GTIN or UPC is
773 the primary feature of the IBM Food Trust. The app also provides end-to-end
774 product provenance, real-time location and status, and facilitates rapid product
775 recalls. The IBM Food Trust also provides insights and visibility for the freshness
776 of the product to reduce losses and spoilage. Lastly, the IBM Food Trust provides
777 certifications from the information taken when handling and managing the prod-
778 ucts in the supply chain. The case studies conducted by Tiwari (2020) illustrates
779 the potential of blockchain technology in improving transparency, efficiency, and
780 ethical practices within supply chains.

781 2.11 Chapter Summary

2.11.1 Comparison Table of Related Studies

Study	Technology Used	Focus Area	Application	Key Findings	Limitations
Shamsuzzoha et al. (2023)	Blockchain (hybrid model)	Fish traceability	Mobile app (Tracey) for fishermen and buyers	Demonstrated feasibility for traceability, certification, transparency; high user acceptability among fishermen	Connectivity issues, technology usability, need for proper incentives and education
Cocco & Mannaro (2021)	Blockchain, IoT (RFID, IPFS), NFC, Ethereum	Italian traditional food (Carasau bread)	Agri-food supply chain traceability system	Proposed a model combining IoT and blockchain to enhance traceability and transparency	High complexity; requires integration of multiple technologies (RFID, IPFS, NFC)
Kresna et al. (2017)	IT-based system, RFID, CCTV, GPS	Tuna supply chain	Digital traceability system (IT-based)	Identified weaknesses in traditional paper-based systems; proposed IT layers for comprehensive tracking	Infrastructure requirements (CCTV, GPS, RFID); limited detail on blockchain
Tiwari (2020) - Provenance	Blockchain, smart contracts, SMS	Tuna tracking certification	Provenance system for tuna supply chain	Enhanced transparency, certification compliance; addressed IUU fishing, fraud, human rights abuses	Limited to specific regions and roles within supply chain
Tiwari (2020) - IBM Food Trust	Blockchain, GTIN/UPC identifiers	Food supply chain	Real-time tracking, product recalls	Improved transparency, efficiency; enabled real-time status, freshness insights	Focused on general food products; high dependency on IBM ecosystem

Table 2.1: Comparison of Studies on Technology for Traceability and Supply Chain Management

782 2.11.2 Research Gaps and Problem

783 Given the advanced existing technologies in blockchain-based traceability systems
 784 for agri-food supply chains, significant gaps remain in understanding the user ex-
 785 perience and integration challenges faced by the fishermen. While the study by
 786 Shamsuzzoha et al. (2023) highlighted the feasibility and benefits of the mobile
 787 application, Tracey, they overlooked possible issues related to technology adop-

788 tion, digital literacy, and connectivity.

789 Furthermore, existing technologies have primarily focused on large-scale imple-
790 mentations and theoretical frameworks without adequately considering the prac-
791 tical implications and user experience needed for effective system integration. This
792 study aims to address these gaps by exploring real-world challenges faced by users,
793 especially fishermen, in adopting blockchain technology for traceability.

794 2.11.3 Summary

795 The literature reviewed highlighted the critical challenges and opportunities re-
796 garding the tuna supply chain, particularly in the areas of traceability and sustain-
797 ability. Existing supply chain technologies, especially those utilizing blockchain,
798 present solutions but also come with limitations in terms of blockchain adoption.
799 The application of blockchain technology in the tuna supply chain has shown po-
800 tential for enhancing traceability from ocean to consumer.

801 Through this paper, a blockchain-driven solution could contribute to providing a
802 more efficient and transparent supply chain. However, further studies are neces-
803 sary to assess the long-term sustainability of blockchain in such systems.

⁸⁰⁴ **Chapter 3**

⁸⁰⁵ **Research Methodology**

⁸⁰⁶ This chapter outlines a clear and detailed description of the research methods and
⁸⁰⁷ processes used in the development and evaluation of SeaXChange: A Blockchain
⁸⁰⁸ Driven App for Tuna Supply Chain Management. The algorithms, systems, theo-
⁸⁰⁹ ries, framework and models are described in detail in which this chapter establishes
⁸¹⁰ the foundation of this study .This chapter also explains the data collection method
⁸¹¹ used ensuring the validity and reliability of the results.In addition, the chapter
⁸¹² discusses the considerations and potential limitations of this study. Overall, this
⁸¹³ will serve as a guide for the readers in understanding the structured process of
⁸¹⁴ developing the SeaXChange.

⁸¹⁵ **3.1 Software Development**

⁸¹⁶ Scrum is a framework within the Agile development that prioritizes flexibility.
⁸¹⁷ It is an iterative software development approach that lets a project be broken

818 down into phases and emphasizes continuous improvements. For this study, the
819 researchers opted to use Scrum because it involved many stakeholders and it
820 operated in a ever-changing environment. Scrum allowed the team to adapt to
821 new requirements through structured sprint planning, weekly reports, and sprint
822 reviews, ensuring continuous alignment with project goals.

823 3.2 Research Activities

824 For this study, the researchers opted for interviews because it enabled in-depth
825 exploration of stakeholder perspectives and experiences. The identified fisher and
826 supplier client interface was tested within the perimeters of Barangay Sapa, Miagao,
827 Iloilo, Philippines. The identified retailer testers were the vendors who reside
828 in Barangay Mat-y and Barangay Sapa in Miagao. The identified consumer testers
829 were situated in Miagao. The researchers chose the local fishermen, retailers and
830 suppliers within Miagao because they wanted to focus on the small-scale fishing.

831 3.2.1 Feedback Collection Method

832 The feedback was collected through a combination of structured interviews and
833 assessment forms. Participants were asked to complete an assessment rubric evalu-
834 ating the SeaXChange app across key criteria such as functionality, end-user needs,
835 performance, usability, ease of use and feasibility. Moreover, follow-up interviews
836 were conducted to gather deeper qualitative insights and obtain suggestions for
837 system improvement.

838 The feedback gathered from fishermen, suppliers and retailers, and consumers
839 was analyzed based on the SeaXChange assessment rubric, which evaluated six
840 major categories: Functionality, End-user Needs, Performance, Usability, Ease of
841 Use and Feasibility. The collected data were analyzed using descriptive statistics,
842 through the computation of mean scores for each assessment criterion. These
843 mean values were used to summarize stakeholder perceptions of the system. Mean
844 ratings were calculated based on the 1-5 Likert Scale where 1 = Poor and 5 =
845 Very Good.

846 3.2.2 Data Gathering

847 • Primary Data:

- 848 – Stakeholder(Fishermen, Supplier, Retailers, and Consumers) interviews
849 were conducted to identify the use-case and user requirements, interface
850 usability, and adoption challenges.
- 851 – Observations were made of existing tuna supply chain processes in local
852 settings.

853 • Secondary Data:

- 854 – Literature review was conducted on blockchain applications in supply
855 chain management and product traceability.
- 856 – Industry reports and regulatory documents related to tuna fishing and
857 supply chain operations.

858 3.2.3 Designing and Developing the System

859 1. **Software Development Methodology:** The project followed a Scrum
860 framework to ensure continuous iteration, stakeholder involvement, and flex-
861 ibility in adapting to feedback.

862 2. **Technology Stack:**

863 • Front-end Development: Used React for creating a secure and user-
864 friendly interface for stakeholders, prioritizing simple and responsive
865 user-interface.

866 • Back-end Development: Used Node.js along with Express for managing
867 back-end processes and API integration. Express is a flexible web applica-
868 tion framework for Node.js used to build APIs for web applications.
869 Docker for containerization of the project and Windows Subsystem for
870 Linux (Ubuntu as the Linux distribution) for setting up the network.

871 • Cloud Infrastructure: Used Google Cloud to host backend services and
872 manage the databases, where the app could be accessed globally. It
873 also ensured the app could scale smoothly as more data and users were
874 added.

875 • Blockchain Framework: Used Go language for developing smart con-
876 tracts and providing an immutable ledger for transaction data.

877 • Database for Accounts: Used Firebase managing user accounts and
878 authentication.

879 3. **Blockchain Development Platform:**

- 880 • Used Hyperledger Fabric for its permissioned nature and scalable ar-
881 chitecture.
- 882 • The open-sourced resources and timely updates of Hyperledger Fabric
883 components is ideal for creating a distributed ledger for tuna supply
884 chain.

885 **3.2.4 Implementing Algorithms and Services**

886 The system for this study is built on top of a Hyperledger Fabric project, it also
887 utilized combinations of algorithms to facilitate the work flow of data or asset
888 as well as ensuring high security with encryption and decryption configuration
889 techniques.

890 **1. Consensus Algorithm**

891 The project followed Raft(Leader-based consensus) for handling organiza-
892 tions or nodes. Raft is intended for managing a replicated log in a blockchain
893 network. Raft is a Crash Fault Tolerant (CFT) protocol, is designed to han-
894 dle non-malicious node failures (e.g., hardware crashes, network issues) In
895 Raft, one node is elected as the leader, and it coordinates the ordering of
896 transactions (Xu et al, 2022) . The leader replicates log entries (transac-
897 tions) to follower nodes, ensuring consistency across the network.

898 **2. Cryptographic Algorithm**

899 The project employed several cryptographic algorithms to ensure security
900 and privacy. These cryptographic data served as digital signatures and iden-
901 tity verification for the project. ECDSA (Elliptic Curve Digital Signature

Algorithm) was used for generating digital signatures while X.509 certificates are intended for identity management and authentication of participants (Anitha & Sankarasubramanian, n.d.) . For the encryption, AES (Advanced Encryption Standard) was used for encrypting data at rest and in transit. TLS (Transport Layer Security) secured communication between network nodes. SHA-256 (Secure Hash Algorithm-256) ensured data integrity by generating unique hashes for blocks and transactions.

3. Membership Service

The implementation of the Membership Service Provider (MSP) requirement involved a set of folders added to the network configuration. These folders defined an organization both internally, by specifying its administrators, and externally, by enabling other organizations to verify the authority of entities attempting specific actions. While Certificate Authorities (CAs) are responsible for generating the certificates that represent identities, the MSP included a list of permitted identities. The MSP specified which Root CAs and Intermediate CAs are authorized to define members of a trust domain. This was achieved by either listing the identities of their members or identifying the CAs allowed to issue valid identities for those members.

4. Ordering Service

The ordering service in this study played a crucial role in maintaining the integrity and functionality of the blockchain network. Its primary responsibilities included ensuring that transactions are processed in the correct sequence (transaction ordering), grouping transactions into blocks based on configurable parameters like size or timeout (block creation), and distributing these ordered blocks to peers for validation and commitment (block

927 distribution) (Nassar et al, 2024). Additionally, the ordering service pro-
928 vided fault tolerance to ensure the network remains operational even in the
929 presence of node failures through Raft.

930 **5. Endorsement Policy**

931 Fabric employs endorsement policies to specify which peers must validate a
932 transaction before it's committed. The algorithm involved multi-signature
933 schemes where a transaction is valid if it receives endorsements from the
934 required peers as per the policy.

935 **6. Chaincode (Smart Contract)**

936 The handling and flow of business logic agreed to by members of the tuna
937 supply chain in the blockchain network is executed by a chaincode or smart
938 contract. The chaincode of the app was written in Go language. Docker
939 container was used for enabling the chaincode to securely run along with the
940 overall hyperledger fabric configurations. Chaincode initializes and manages
941 ledger state through transactions submitted by applications (Hyperledger
942 Fabric Documentation, 2024) . The chaincode followed the object-oriented
943 paradigm for creating classes and objects necessary for the tuna supply
944 chain.

945 **3.2.5 Modeling the System Architecture**

946 The system architecture of the project were consisted of many nodes that com-
947 municated with each other. The chaincode enabled the system to run algorithms,
948 particularly, holding state and ledger data, and executes transactions such as asset
949 transfer in the tuna supply chain.

950 • **Blockchain Architecture**

951 The project involved peer, ordering services, ledger, and client application
 952 to perform various transaction such as tracing the origin and the stop points
 953 of a tuna asset. Peers are nodes in the blockchain network that maintained
 954 a copy of the distributed ledger and execute chaincode (smart contracts).
 955 The ordering service is the central component of the blockchain for ordering
 956 transactions and creating blocks to distribute to peers through consensus
 957 mechanism. The ledger is the immutable record of all transaction in the
 958 tuna supply chain network, stored across all peers. The client application is
 959 the interface through which users or tuna supply chain participants interact
 960 with the blockchain network.

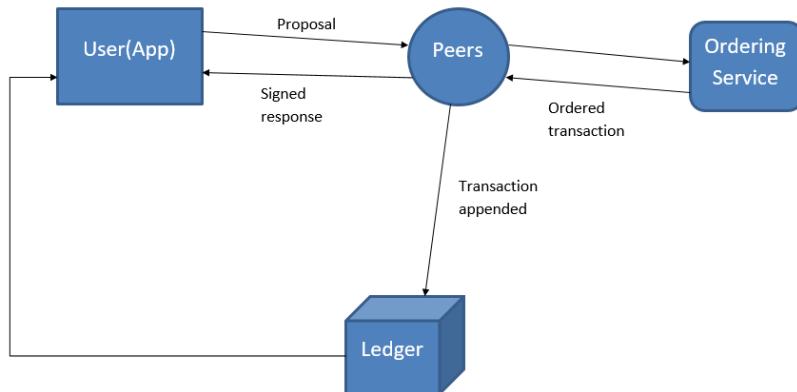


Figure 3.1: Blockchain Architecture of SeaXChange

961 • **Overall System Architecture**

962 The overall system architecture contains a web application built with Next.js
 963 for the frontend, utilizing Firebase for user authentication and account man-
 964 agement. The application follows a role-based access model (Fisher, Sup-
 965 plier, Retailer, Consumer) where each role has specific permissions for inter-

966 acting with tuna assets in the supply chain. The backend runs on Google
 967 Cloud Platform, consisting of an Express.js API that interfaces with a Hyperledger Fabric blockchain network (containerized in Docker) which stores
 968 and manages the immutable record of tuna assets and their transfers between supply chain participants. This architecture enables secure tracking
 969 of tuna from creation by fishers through the supply chain to consumers, with appropriate viewing and transfer capabilities assigned to each role in
 970 the process.
 971
 972
 973

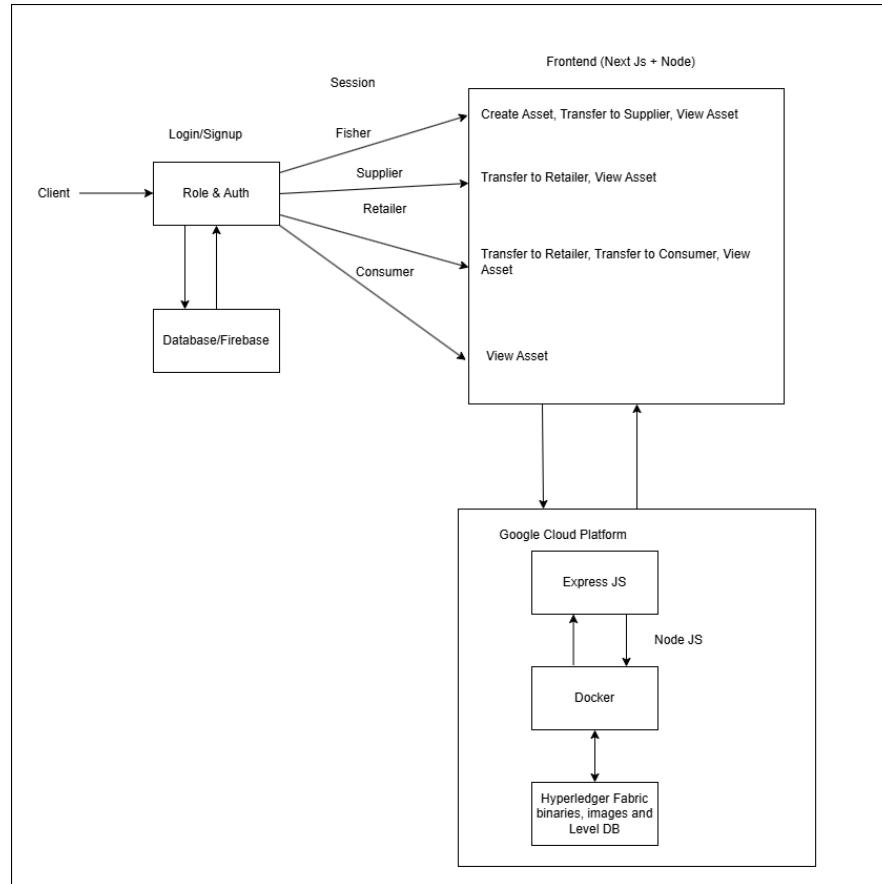


Figure 3.2: Overall System Architecture of SeaXChange

975 The use case shows the outline on how the user will interact with the SeaX-
976 Change App. It followed the major stages or participants in the tuna supply
977 chain.

978 **1. Fisher**

- 979 - Encodes tuna I.D. of fish.
980 - Encodes the date when the fish was captured.
981 - Encodes the location where the fish was captured.
982 - Encodes the fishing method used.
983 - Query the origin and exchange of the tuna asset.

984 **2. Supplier**

- 985 - Encodes when the product was transferred from fisher to supplier.
986 - Query the origin and exchange of the tuna asset.
987 - Generate supplier's location during retrieval of tuna asset.

988 **3. Retailer**

- 989 - Encodes when the product was retrieved from the supplier or another
990 retailer.
991 - Query the origin and exchange of the tuna asset.
992 - Generate retailer's location during retrieval of tuna asset.

993 **4. Consumer**

- 994 - Retrieve data from retailer.
995 - Query the origin and exchange of the tuna asset.

996 There are four (4) types of users that will use the app. The first user
997 type is the Fisher, which will be the starting point of the blockchain.
998 It will encode the catch details of a tuna product such as the date of
999 capture, location, and fishing method. The second user type is the

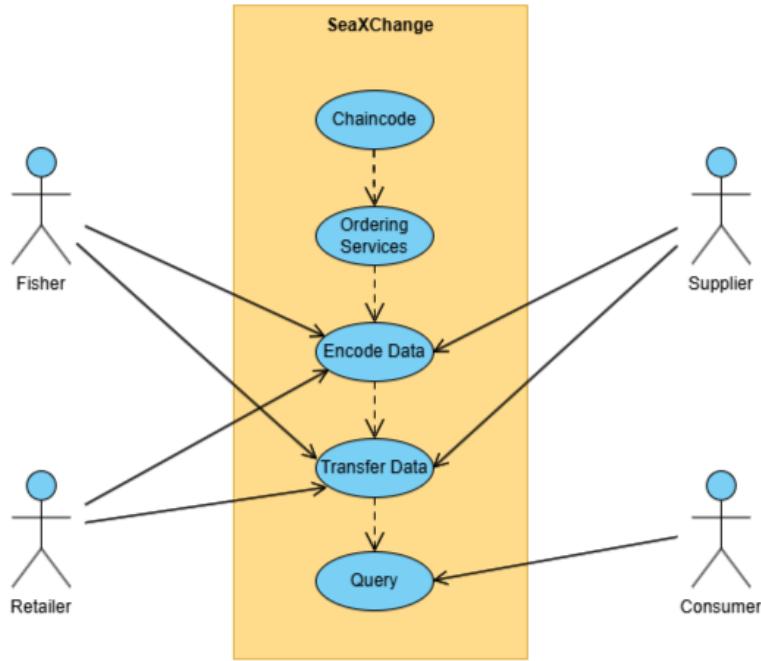


Figure 3.3: Use case diagram for SeaXChange.

1000 Supplier, which will encode when the product was transferred from
 1001 the fisher to the supplier, as well as generate their location during
 1002 the retrieval of the tuna asset. The third type is the Retailer, which
 1003 will encode when the product was transferred from the supplier to the
 1004 retailer or in the case of multiple retailers, from the previous retailer to
 1005 the current retailer, their location is also generated during the retrieval
 1006 of the tuna asset. Lastly, the Consumers, which can only query the
 1007 origin and exchange of tuna assets.

¹⁰⁰⁸ **Chapter 4**

¹⁰⁰⁹ **Results and Discussions**

¹⁰¹⁰ **4.1 Overview**

¹⁰¹¹ This chapter presents the results of the system. Included in this chapter are
¹⁰¹² screenshots and the discussion of results. The tuna supply chain management
¹⁰¹³ smart contract on Hyperledger Fabric has been initiated and tested within a con-
¹⁰¹⁴ trolled blockchain environment. Results indicated that the system was function-
¹⁰¹⁵ ally robust and reliable, having managed assets, transaction integrity, and the
¹⁰¹⁶ ability to query and update the ledger in the blockchain. This chapter presents
¹⁰¹⁷ the details of the major steps executed during the process, results for those steps,
¹⁰¹⁸ and the current status of the system's operations.

1019 **4.2 Smart Contract Deployment and Installa-**

1020 **tion**

1021 **4.2.1 Hyperledger Fabric Prerequisites**

1022 Before executing a smart contract framework and blockchain system, it is cru-
1023 cial to first install and set up the necessary tools and technologies. This includes
1024 setting up Hyperledger Fabric, which involves installing the Fabric binaries, con-
1025figuring the network, and ensuring all necessary dependencies like Docker, Docker
1026 Compose, and Node.js are installed and properly configured. Additionally, setting
1027 up the required certificates, defining the channel configurations, and ensuring that
1028 peer nodes and orderers are correctly connected and synchronized are all essential
1029 steps in preparing the environment for blockchain and smart contract operations.

1030 • **Software Requirements:**

- 1031 – **Docker and Docker Compose:** Hyperledger Fabric needs to have
1032 Docker installed and running on the system. Docker is required to run
1033 the peer and ordering services of the blockchain system.
- 1034 – **Node.js:** Required for the Fabric SDK for client application integra-
1035 tion with JavaScript libraries such as react.
- 1036 – **Go:** Ensure Go is installed, and the GOPATH environment variable
1037 is set up. This is essential for building and running chaincode(smart
1038 contract) written in Go.
- 1039 – **Fabric Samples:** Clone the official Hyperledger Fabric's fabric-samples
1040 repository from GitHub:

```
1041     git clone -b release-2.4 --single-branch  
1042         https://github.com/hyperledger/fabric-samples  
1043     cd fabric-samples/test-network  
1044
```

1045 – **Binaries and Docker Images:**

```
1046 curl -sSL https://bit.ly/2ysbOFE | bash -s
```

```
1047
```

1048 • **Network Setup:**

1049 – Run the `test-network` script to start the Hyperledger Fabric test net-
1050 work:

```
1051     ./network.sh up
```

```
1052
```

1053 This script starts a peer, an ordering service, and a CA (Certificate
1054 Authority) on the local machine.

1055 – After starting the network to docker in the same directory (`test-network`),
1056 a channel must be created:

```
1057     ./network.sh createChannel
```

```
1058
```

1059 • **Deploying Chaincode (Smart Contract):**

1060 – Step 1:

```
1061     export PATH=${PWD}/../bin:$PATH
```

```
1062
```

1063 – Step 2:

```
1064     export FABRIC_CFG_PATH=$PWD/./config/
```

```
1065
```

1066 – Step 3:

```
1067     export CORE_PEER_TLS_ENABLED=true
1068     export CORE_PEER_LOCALMSPID="Org1MSP"
1069     export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations
1070         /peerOrganizations/org1.example.com/peers/peer0.org1.example.com
1071             /tls/ca.crt
1072     export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations
1073         /peerOrganizations/org1.example.com/users
1074             /Admin@org1.example.com/msp
1075     export CORE_PEER_ADDRESS=localhost:7051
1076
1077
```

1078 4.2.2 Invoking the Blockchain System

1079 After setting up the prerequisites, including Docker containers, the test network,
1080 and chaincode, the tuna supply chain system can now be invoked for creating,
1081 transferring, and querying tuna assets. The figures provided below demonstrate
1082 the processes involved in invoking the blockchain system.

4.2. SMART CONTRACT DEPLOYMENT AND INSTALLATION

45

```
[root@ryrex2002@LAPTOP-Q93UQUB8:/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-network]# peer chaincode query -C mychannel -n basic_cc -f "queryAllAssets"
[{"ID": "tuna1", "Species": "Skipjack", "Weight": 5.5, "CatchLocation": "Antique", "CatchDate": "2024-12-01", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierX", "Retailer": "Reyes"}, {"ID": "tuna2", "Species": "Yellowfin", "Weight": 8.5, "CatchLocation": "Palawan", "CatchDate": "2024-12-02", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierY", "Retailer": "Cruz"}, {"ID": "tuna3", "Species": "Bluefin", "Weight": 5.5, "CatchLocation": "Antique", "CatchDate": "2024-12-03", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierZ", "Retailer": "Cruz"}, {"ID": "tuna4", "Species": "Skipjack", "Weight": 5.5, "CatchLocation": "Antique", "CatchDate": "2024-12-04", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierX", "Retailer": "MA"}, {"ID": "tuna5", "Species": "Albacore", "Weight": 6, "CatchLocation": "Davao", "CatchDate": "2024-12-05", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierA", "Retailer": "MA"}, {"ID": "tuna6", "Species": "Skipjack", "Weight": 6, "CatchLocation": "Davao", "CatchDate": "2024-12-06", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierB", "Retailer": "MA"}]
[ryrex2002@LAPTOP-Q93UQUB8:/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-network]
```

Figure 4.1: Query Smart Contract: Check assets

1083 • Adding new tuna assets:

1084 Here, a new tuna asset is created and registered on the blockchain. This in-
 1085 volves invoking the smart contract to add a new entry, which includes details
 1086 such as the type of tuna, quantity, and any other relevant information. This
 1087 step ensures that newly caught or acquired tuna can be tracked throughout
 1088 the supply chain.

```
[root@ryrex2002@LAPTOP-Q93UQUB8:/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-network]# peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tlsRootCertFiles $PWD/organizations/ordererOrganizations/example.com/msp/tlscacerts/tlscacert.pem -C mychannel -n basic_cc -c '{"function": "CreateAsset", "Args": [{"ID": "tuna7", "Species": "Skipjack", "Weight": 6, "CatchLocation": "Antique", "CatchDate": "2024-12-07", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierX", "Retailer": "MA"}]}
[ryrex2002@LAPTOP-Q93UQUB8:/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-network]
```

Figure 4.2: Invoke Smart Contract: Create/Add new tuna asset

1089 • Check all assets after adding a new tuna asset:

1090 After adding a new tuna asset, the smart contract is queried again to verify
 1091 that the asset has been successfully added. This step confirms that the new
 1092 asset is part of the current inventory and that no discrepancies exist in the
 1093 recorded data.

```
[root@ryrex2002@LAPTOP-Q93UQUB8:/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-network]# peer chaincode query -C mychannel -n basic_cc -f "queryAllAssets"
[{"ID": "tuna1", "Species": "Skipjack", "Weight": 5.5, "CatchLocation": "Antique", "CatchDate": "2024-12-01", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierX", "Retailer": "Reyes"}, {"ID": "tuna2", "Species": "Yellowfin", "Weight": 8.5, "CatchLocation": "Palawan", "CatchDate": "2024-12-02", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierY", "Retailer": "Cruz"}, {"ID": "tuna3", "Species": "Bluefin", "Weight": 5.5, "CatchLocation": "Antique", "CatchDate": "2024-12-03", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierZ", "Retailer": "Cruz"}, {"ID": "tuna4", "Species": "Skipjack", "Weight": 5.5, "CatchLocation": "Antique", "CatchDate": "2024-12-04", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierX", "Retailer": "MA"}, {"ID": "tuna5", "Species": "Albacore", "Weight": 6, "CatchLocation": "Davao", "CatchDate": "2024-12-05", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierA", "Retailer": "MA"}, {"ID": "tuna6", "Species": "Skipjack", "Weight": 6, "CatchLocation": "Davao", "CatchDate": "2024-12-06", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierB", "Retailer": "MA"}, {"ID": "tuna7", "Species": "Skipjack", "Weight": 6, "CatchLocation": "Antique", "CatchDate": "2024-12-07", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierX", "Retailer": "MA"}]
[ryrex2002@LAPTOP-Q93UQUB8:/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-network]
```

Figure 4.3: Query Smart Contract: Check assets with new tuna asset

1094 • Transfer tuna asset to Supplier:

1095 This step involves transferring ownership of a tuna asset from the current

holder (e.g., a fisherman or a trader) to a supplier. The smart contract is invoked to facilitate the transfer, ensuring that the transaction is securely recorded on the blockchain and that the asset's new owner is updated accordingly.

```
1096      Error: error getting broadcast client: orderer client failed to connect to orderer.example.com:7050: failed to create new connection: context deadline exceeded
1097      rsys2002@LAPTOP-Q93QUN8:/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-network$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tlsCafile ./msp/tlscacerts/tlscacert.pem --peerChannelName basic-peer --peerAddresses localhost:9051 --tlsRootCertFiles ./msp/tlscacerts/tlscacert.pem -C "function" "transferAsset" "A
1098      rs":["tuna6","Supplier","SupplierA"]
1099      rs":["tuna6","Supplier","SupplierA"]'
```

Figure 4.4: Invoke Smart Contract: Transfer asset to a supplier

1100 • Check the updated tuna asset:

After the transfer, the smart contract is queried once more to check if the asset details have been updated correctly. This step verifies that the asset's new owner is now the supplier and that all relevant information is correctly updated on the blockchain.

```
1101      rsys2002@LAPTOP-Q93QUN8:/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-
1102      network$ peer chaincode query -C mychannel -n basic -c '{"Args":["TransferAsset","tuna6","Supplier","Jose
1103      rys","SupplierA"]}' --ordererTLSHostnameOverride orderer.example.com:7050
1104      Error: error getting broadcast client: orderer client failed to connect to orderer.example.com:7050: failed to create new connection: context deadline exceeded
1105      rsys2002@LAPTOP-Q93QUN8:/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-network$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tlsCafile ./msp/tlscacerts/tlscacert.pem --peerChannelName basic-peer --peerAddresses localhost:9051 --tlsRootCertFiles ./msp/tlscacerts/tlscacert.pem -C "function" "transferAsset" "A
1106      rs":["tuna6","Supplier","SupplierA"]'
```

Figure 4.5: Query Smart Contract: Check asset after transfer

1105 • Transfer tuna asset to Retailer:

Similar to the supplier transfer, this step involves transferring the tuna asset from the supplier to a retailer. The smart contract facilitates this transfer, ensuring that ownership is correctly updated and that the retailer has control over the tuna asset. This step is crucial for the supply chain as it moves the tuna from bulk supply to retail.

```
1106      [{"ID":"tuna6","Species":"Skipjack","Weight":6,"CatchLocation":"2024-12-05T00:00:00Z","CatchDate":"Antique","FishingMethod":"Longline","Vessel":"Jagney","Supplier":"SupplierA","Retailer":"RetailerA"}]
1107      rsys2002@LAPTOP-Q93QUN8:/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-network$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tlsCafile ./msp/tlscacerts/tlscacert.pem --peerChannelName basic-peer --peerAddresses localhost:9051 --tlsRootCertFiles ./msp/tlscacerts/tlscacert.pem -C "function" "transferAsset" "A" "rs":["tuna6","Supplier","SupplierA"]'
1108      rs":["tuna6","Supplier","SupplierA"]'
```

Figure 4.6: Invoke Smart Contract: Transfer asset to a retailer

4.3. BACKEND SECURITY ANALYSIS (HYPERLEDGER FABRIC ON GCP)47

1111 • **Check the updated tuna asset:**

1112 After the transfer to the retailer, another query is made to verify the updated
1113 asset details. This step ensures that the transaction was successful and that
1114 the retailer now has ownership of the tuna asset. It confirms that the asset
1115 has moved through the supply chain correctly.

```
zyrex2082@LAPTOP-Q93UQUB8:/mnt/c/SpecialProblem/hyperledger-fabric3/Fabric-samples/test-network$ peer chaincode query -C mychannel -n basic -c '{"Args":["ReadAsset","tuna6"]}'  
[{"ID": "tuna6", "Species": "Skipjack", "Weight": 6.5, "CatchLocation": "Antique", "CatchRate": "200-12-01", "FishingMethod": "Longline", "Vessel": "Jagnee", "Supplier": "SupplierA", "Retailer": "Jose"}]  
zyrex2082@LAPTOP-Q93UQUB8:/mnt/c/SpecialProblem/hyperledger-fabric3/Fabric-samples/test-network$
```

Figure 4.7: Query Smart Contract: Check asset after transfer

1116 • **Query Smart Contract and check updated assets:**

1117 The final step involves querying the smart contract to get a complete overview
1118 of all the assets in the supply chain. This includes all tuna assets from fishing
1119 to retail, allowing stakeholders to monitor and manage inventory effectively.
1120 It provides traceability in the supply chain, helping to maintain freshness
1121 and authenticity of the tuna as it moves through the market.

```
zyrex2082@LAPTOP-Q93UQUB8:/mnt/c/SpecialProblem/hyperledger-fabric3/Fabric-samples/test-network$ peer chaincode query -C mychannel -n basic -c '{"Args":["getAllAssets"]}'  
[{"ID": "tuna1", "Species": "Skipjack", "Weight": 6.5, "CatchLocation": "Antique", "CatchRate": "200-12-01", "FishingMethod": "Longline", "Vessel": "Jagnee", "Supplier": "SupplierA", "Retailer": "Reyes"}, {"ID": "tuna2", "Species": "Yellowfin", "Weight": 8.5, "CatchLocation": "Palawan", "CatchDate": "2024-12-02", "FishingMethod": "Longline", "Vessel": "Jagnee", "Supplier": "SupplierA", "Retailer": "Reyes"}, {"ID": "tuna3", "Species": "Bluefin", "Weight": 5.5, "CatchLocation": "Antique", "CatchDate": "2024-12-03", "FishingMethod": "Longline", "Vessel": "Jagnee", "Supplier": "SupplierA", "Retailer": "Santos"}, {"ID": "tuna4", "Species": "Bull", "Weight": 5.5, "CatchLocation": "Antique", "CatchDate": "2024-12-03", "FishingMethod": "Longline", "Vessel": "Jagnee", "Supplier": "SupplierA", "Retailer": "Santos"}, {"ID": "tuna5", "Species": "Albacore", "Weight": 6, "CatchLocation": "2024-12-05", "CatchDate": "2024-12-05", "FishingMethod": "Longline", "Vessel": "Jagnee", "Supplier": "SupplierA", "Retailer": "Mia"}, {"ID": "tuna6", "Species": "Skipjack", "Weight": 6, "CatchLocation": "2024-12-05", "CatchDate": "2024-12-05", "FishingMethod": "Longline", "Vessel": "Jagnee", "Supplier": "SupplierA", "Retailer": "Mia"}]  
zyrex2082@LAPTOP-Q93UQUB8:/mnt/c/SpecialProblem/hyperledger-fabric3/Fabric-samples/test-network$
```

Figure 4.8: Query Smart Contract: Check updated assets

₁₁₂₂ **4.3 Backend Security Analysis (Hyperledger Fab-**

₁₁₂₃ **ric on GCP)**

₁₁₂₄ **4.3.1 System Architecture and Deployment Overview**

₁₁₂₅ The backend of the system's tuna assets was developed using a containerized
₁₁₂₆ Hyperledger Fabric deployed on Google Cloud Platform (GCP). The network of
₁₁₂₇ Hyperledger Fabric consists of a peer node, an ordering node, and Certificate
₁₁₂₈ Authorities (CAs).

₁₁₂₉ **4.3.2 Blockchain Network Security**

₁₁₃₀ The blockchain network leverages Hyperledger Fabric's security model to ensure
₁₁₃₁ authenticated transactions and controlled access. A Membership Service Provider
₁₁₃₂ (MSP) manages identities and issues certificates based on a Public Key Infrastruc-
₁₁₃₃ ture (PKI) model, ensuring that only verified participants can interact with the
₁₁₃₄ network.

₁₁₃₅ Key security features include:

₁₁₃₆ **Channel Privacy**

₁₁₃₇ Channels act as private communication subnets, isolating transaction data so that
₁₁₃₈ only authorized organizations can access and process specific transactions.

4.3. BACKEND SECURITY ANALYSIS (HYPERLEDGER FABRIC ON GCP)49

1139 Policies and Access Control

1140 Policies, including endorsement policies and access control lists (ACLs), govern
1141 how transactions are validated, how channel resources are accessed, and how
1142 changes to the network are approved. Endorsement policies specifically define
1143 which peer nodes must approve a transaction before it is committed to the ledger.

1144 Secure Communication

1145 Transport Layer Security (TLS) is enforced across node communications to protect
1146 data in transit. Mutual TLS is used for operational endpoints like monitoring
1147 services.

1148 Identity and Role Management

1149 Every network participant—peer nodes, orderer nodes, client applications (SeaX-
1150 Change Web Application)—has a cryptographically verifiable identity, with roles
1151 defined within the framework to control access and permissions within channels.

1152 Hardware Security Modules (HSMs)

1153 Critical cryptographic operations, such as signing transactions under the blockchain
1154 assets invocation, can optionally be handled by HSMs to secure private keys out-
1155 side of the software environment.

1156 These layered mechanisms collectively ensure the confidentiality, integrity, and

1157 authenticity of transactions in the blockchain network.

1158 4.3.3 Smart Contract Automated Test Result

1159 To validate the security and functionality of the deployed smart contracts on the
1160 Hyperledger Fabric network, an automated testing script (app.js) under asset-
1161 transfer-basic directory was executed. The script interacted with the blockchain
1162 network through the gateway application, utilizing the defined channel (mychan-
1163 nel) and chaincode (basic). The automated tests performed the following opera-
1164 tions:

1165 InitLedger Transaction

1166 The ledger was initialized by creating a predefined set of tuna asset entries. The
1167 transaction was successfully committed, confirming the proper initialization of
1168 asset data. (See Figure 4.9 for initialization confirmation.)

```
cd /opt/gopath/src/github.com/SpecialProblem/hyperledger-fabric-samples/asset-transfer-basic/application-gateway-javascript/src; node app.js
channelName: mychannel
chaincodeName: basic
msp: Org1MSP
cryptoPath: /mnt/c/SpecialProblem/hyperledger-fabric3/Fabric-samples/test-network/organizations/peerOrganizations/org1.example.com
keyDirectoryPath: /mnt/c/SpecialProblem/hyperledger-fabric3/Fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/users/User1@org1.example.com/msp
certDirectoryPath: /mnt/c/SpecialProblem/hyperledger-fabric3/Fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/users/User1@org1.example.com/msp
signcerts
tlsCertPath: /mnt/c/SpecialProblem/hyperledger-fabric3/Fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls
caCertPath: /mnt/c/SpecialProblem/hyperledger-fabric3/Fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/ca/tls
peerEndpoint: localhost:7051
peerHostAlias: peer0.org1.example.com
--> Submit Transaction: InitLedger, function creates the initial set of assets on the ledger
*** Transaction committed successfully
```

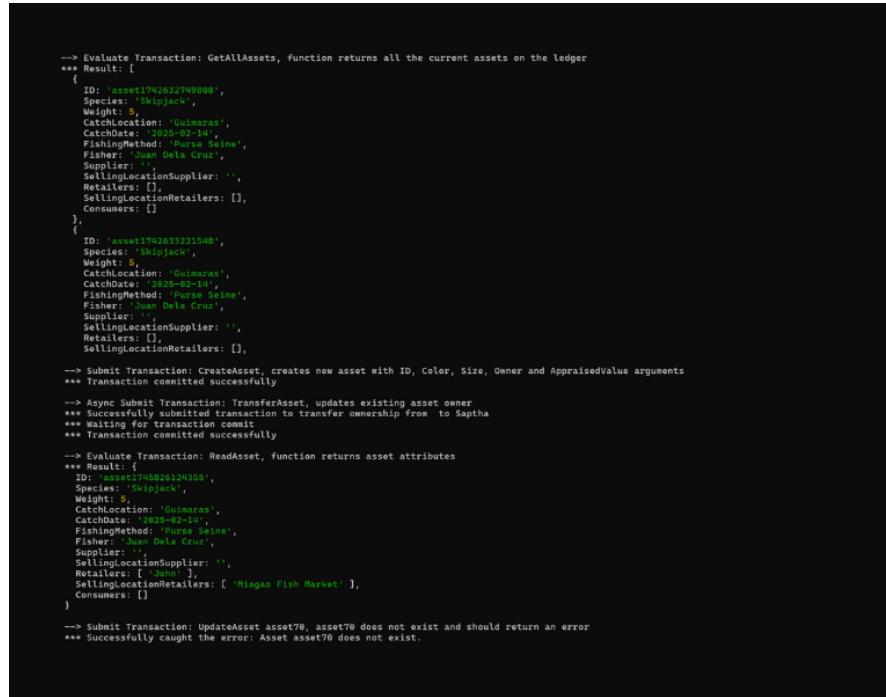
Figure 4.9: Initialization Confirmation of the Ledger

1169 GetAllAssets Query

1170 A query operation retrieved all existing assets recorded on the ledger. The results
1171 displayed multiple tuna asset entries with details such as species, weight, catch

4.3. BACKEND SECURITY ANALYSIS (HYPERLEDGER FABRIC ON GCP)51

1172 location, catch date, fishing method, fisher, supplier, retailers, selling locations,
1173 and consumers. (See Figure 4.10 for the asset retrieval output.)



```
--> Evaluate Transaction: GetAllAssets, function returns all the current assets on the ledger
*** Result: [
  {
    ID: 'asset174632749008',
    Species: 'Skipjack',
    Weight: 5,
    CatchLocation: 'Guimaras',
    CatchDate: '2025-02-14',
    FishingMethod: 'Purse Seine',
    Fisher: 'Juan Dela Cruz',
    Supplier: '',
    SellinglocationSupplier: '',
    Retailers: [],
    SellinglocationRetailers: [],
    Consumers: []
  },
  {
    ID: 'asset174633221548',
    Species: 'Skipjack',
    Weight: 5,
    CatchLocation: 'Guimaras',
    CatchDate: '2025-02-14',
    FishingMethod: 'Purse Seine',
    Fisher: 'Juan Dela Cruz',
    Supplier: '',
    SellinglocationSupplier: '',
    Retailers: [],
    SellinglocationRetailers: []
  }
]
--> Submit Transaction: CreateAsset, creates new asset with ID, Color, Size, Owner and AppraisedValue arguments
*** Transaction committed successfully
--> Asynchronous Submit Transaction: TransferAsset, updates existing asset owner
*** Successfully submitted transaction to transfer ownership from to Saptha
*** Waiting for transaction commit
*** Transaction committed successfully
--> Evaluate Transaction: ReadAsset, function returns asset attributes
*** Result: [
  {
    ID: 'asset174636124555',
    Species: 'Skipjack',
    Weight: 5,
    CatchLocation: 'Guimaras',
    CatchDate: '2025-02-14',
    FishingMethod: 'Purse Seine',
    Fisher: 'Juan Dela Cruz',
    Supplier: '',
    SellinglocationSupplier: '',
    Retailers: [ 'John' ],
    SellinglocationRetailers: [ 'Miagao Fish Market' ],
    Consumers: []
  }
]
--> Submit Transaction: UpdateAsset asset70, asset70 does not exist and should return an error
*** Successfully caught the error: Asset asset70 does not exist.
```

Figure 4.10: Initialization Confirmation of the Ledger

1174 CreateAsset Transaction

1175 A new asset was successfully created and appended to the blockchain. The trans-
1176 action was committed without errors, validating the chaincode's ability to handle
1177 new data insertion.

1178 TransferAsset Transaction

1179 Ownership transfer of an existing asset was simulated. The transaction was suc-
1180 cessfully submitted and committed, demonstrating the correct application of asset

1181 updates in the blockchain ledger.

1182 **ReadAsset Query**

1183 The updated asset was retrieved and verified to ensure the correctness of the
1184 transfer. The retrieved asset data reflected the changes made through the previous
1185 transaction, confirming data consistency.

1186 **UpdateAsset Error Handling**

1187 An attempt to update a non-existent asset (asset70) was performed to test the
1188 smart contract's error-handling mechanism. The application correctly caught and
1189 reported the error, verifying that improper transactions are adequately handled
1190 and rejected.

1191 **Summary of Results**

1192 All positive transactions (initialization, creation, transfer, and reading) were suc-
1193 cessfully executed and committed. The smart contract exhibited robust error
1194 handling during invalid operations. Endorsement policies and Membership Ser-
1195 vice Provider (MSP) enforcement ensured transaction authenticity and integrity
1196 during the process. These tests confirm the functional reliability and transac-
1197 tion security of the smart contracts used within the tuna supply chain blockchain
1198 network.

¹¹⁹⁹ **4.3.4 GCP Infrastructure Security**

¹²⁰⁰ The Hyperledger Fabric network deployment on Google Cloud Platform (GCP)
¹²⁰¹ was secured by leveraging multiple layers of Google's infrastructure security model,
¹²⁰² following best practices in network, identity, and data protection.

¹²⁰³ **Firewall Rules and Network Control**

¹²⁰⁴ Only essential ports (e.g., 7051 for peer communication and 7050 for the ordering
¹²⁰⁵ service) were opened to minimize network exposure. GCP's VPC firewall rules,
¹²⁰⁶ ingress and egress controls, and options like VPC Service Controls help further
¹²⁰⁷ isolate services and prevent unauthorized access. Traffic between virtual machines
¹²⁰⁸ and Google APIs is securely routed through Google's internal network infrastruc-
¹²⁰⁹ ture using Private Google Access when available.

¹²¹⁰ **IAM Roles and Access Management**

¹²¹¹ The principle of least privilege was enforced by assigning minimal permissions to
¹²¹² users and services through GCP Identity and Access Management (IAM). Access
¹²¹³ decisions involve authentication, authorization checks, and enforcement of policies
¹²¹⁴ through centralized services, helping reduce the risk of unauthorized actions or
¹²¹⁵ privilege escalation.

1216 Encryption

1217 GCP ensures that data is encrypted both at rest and in transit by default. Storage
1218 systems use multiple layers of encryption, with cryptographic keys managed by
1219 Google. For additional control, Cloud Key Management Service (KMS) enables
1220 customers to manage their own encryption keys. Data in transit between services
1221 is protected using Application Layer Transport Security (ALTS), and all external
1222 communication with Google services is encrypted via TLS.

1223 Access Control

1224 Access to virtual machines and services was restricted using secure access methods.
1225 Identity-Aware Proxy (IAP) or VPN connections were employed to safeguard SSH
1226 and administrative access. GCP's zero-trust model emphasizes verifying identity
1227 and device security rather than relying solely on network location, aligning with
1228 best practices for modern infrastructure protection.

1229 Infrastructure and Operational Security

1230 GCP's underlying infrastructure benefits from Google's proprietary hardware de-
1231 signs, including the Titan security chip, secure boot mechanisms, and service iden-
1232 tity enforcement. Google's physical data centers use multi-layered defenses such
1233 as biometrics and intrusion detection systems. Operational security practices, in-
1234 cluding binary authorization and extensive monitoring, reduce insider risks and
1235 enforce software integrity throughout the lifecycle.

₁₂₃₆ By deploying the blockchain network on GCP, the project leveraged a highly se-
₁₂₃₇ cure environment, benefiting from Google's layered security architecture across
₁₂₃₈ networking, identity management, encryption, access control, and operational
₁₂₃₉ practices.

₁₂₄₀ 4.3.5 Threat Model and Mitigations

₁₂₄₁ Potential threats to the system were identified and mitigation strategies were applied, as summarized in Table 4.1.

Threat	Mitigation
Unauthorized access to network	Use of MSP and Certificate Authorities
Tampering with transactions	Endorsement policies and consensus mechanisms
Denial of Service (DoS) attacks	GCP Firewall and rate limiting rules
Data leakage	Private channels and access controls

Table 4.1: Potential Threat and Mitigation

₁₂₄₂

₁₂₄₃ 4.4 Mockups

₁₂₄₄ The mockups represent the preliminary design for the SeaXChange web applica-
₁₂₄₅ tion, created using Figma to facilitate collaboration and incorporate feedback
₁₂₄₆ efficiently. The visual design features a teal-based color scheme to evoke an oceanic
₁₂₄₇ theme, aligning with the app's focus on tuna products supply chain. This aesthetic
₁₂₄₈ choice reinforces the app's identity and enhances user engagement.

₁₂₄₉ Upon launching the app, users are first directed to the Login or Sign-Up page,
₁₂₅₀ where authentication is required to access any data. This ensures security and

1251 role-specific access within the blockchain system.

1252 Once authenticated, users are redirected to their personalized homepages, which
1253 include their profile information and a dashboard showing relevant tuna product
1254 assets. The user experience is role-dependent:

1255 • All users can view available assets on the blockchain.

1256 • Fishermen are the only users who can create new assets, representing newly
1257 caught tuna.

1258 • Suppliers and Retailers have the ability to pass on assets down the supply
1259 chain, updating the product's status, location, or freshness.

1260 This role-based structure ensures traceability, accountability, and clarity across
1261 the supply chain, while maintaining a clean and intuitive interface tailored to
1262 each user type.

4.4. MOCKUPS

57

Login Page

SeaXChange

Email:

Password:

Log In

Don't have an account? [Sign up](#)

Fisher Homepage

SeaXChange

Order Tuna 32

+ ADD CATCH

Picture Picture Picture

TUNA1 Dec 1, 2024 Available

TUNA1 Dec 1, 2024 Available

TUNA1 Dec 1, 2024 Sold

Picture Picture Picture

TUNA1 Dec 1, 2024 Available

TUNA1 Dec 1, 2024 Available

TUNA1 Dec 1, 2024 Available

Fisher Add Catch Page

SeaXChange

TUNA1

Species: Skipjack

Weight (kg): 5.5

Catch Location: Antipe

Catch Date: 2024-10-01

Fishing Method: Longline

Vessel: Japone

Supplier: NA

Retailer: NA

Consumer: NA

SAVE **SEND TUNA**

Fisher Add Catch Page 2

SeaXChange

TUNA1

Species: Skipjack

Weight (kg): 5.5

Catch Location: Antipe

Catch Date: 2024-10-01

Fishing Method: Longline

Vessel: Japone

Supplier: NA

Retailer: NA

Consumer: NA

SEND **EDIT**

SAVE **SEND TUNA**

Supplier Homepage

SeaXChange

Order Tuna 32

Picture Picture Picture

TUNA1 Dec 1, 2024 Available

TUNA1 Dec 1, 2024 Available

TUNA1 Dec 1, 2024 Sold

Picture Picture Picture

TUNA1 Dec 1, 2024 Available

TUNA1 Dec 1, 2024 Available

TUNA1 Dec 1, 2024 Available

Supplier Search/Click Result Page

SeaXChange

TUNA1

Species: Skipjack

Weight (kg): 5.5

Catch Location: Antipe

Catch Date: 2024-10-01

Fishing Method: Longline

Vessel: Japone

Supplier: John Doe

Retailer: NA

Consumer: NA

SEND TUNA

Retailer Search/Sell Page

SeaXChange

TUNA1

Species: Skipjack

Weight (kg): 5.5

Catch Location: Antipe

Catch Date: 2024-10-01

Fishing Method: Longline

Vessel: Japone

Supplier: NA

Retailer: Oscar Gomes

SEND TUNA **MARK AS SOLD**

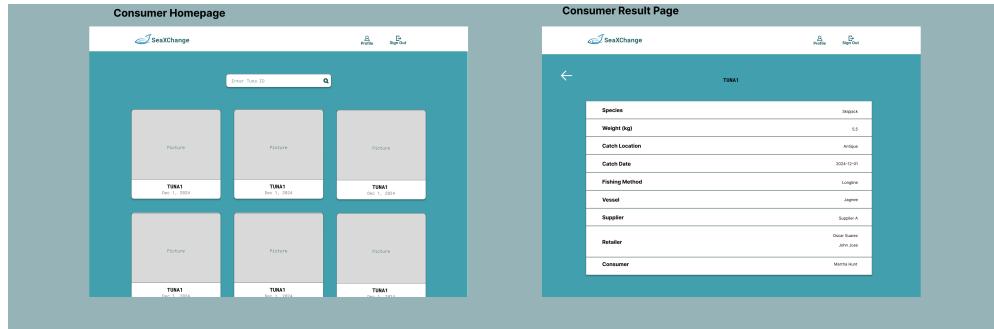


Figure 4.11: SeaXChange Mockups showing the Authentication Page, Role-Based Homepage, Asset Transfer Interfaces for Fishermen, Suppliers, and Retailers, and Asset Viewing page for the Consumers

¹²⁶³ 4.5 Operational Flow of the Web Application

¹²⁶⁴ This section will discuss the flow in using the SeaXChange Web Application. It
¹²⁶⁵ will show the respective interface for every page and how the users can interact
¹²⁶⁶ with it.

¹²⁶⁷ 4.5.1 Landing Page

¹²⁶⁸ Users are be greeted with the landing page, where it shows a ocean visuals and
¹²⁶⁹ a tagline “Discover the Journey your tuna made from the ocean to your dinner
¹²⁷⁰ plate”. Users are given the option to Login, where they are redirected to the login
¹²⁷¹ page or Get Started, where they are redirected to the sign up page.



Figure 4.12: Landing Page

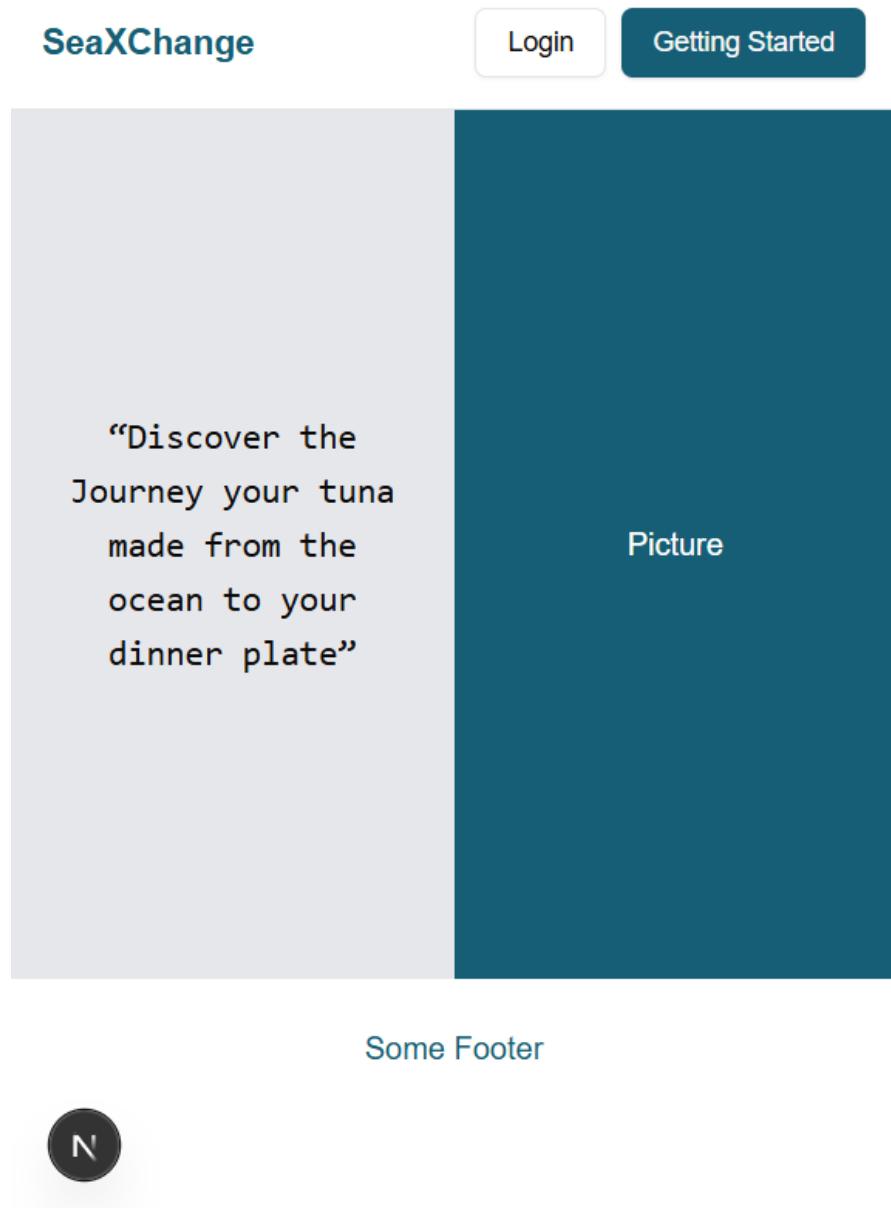


Figure 4.13: Mobile View: Landing Page

1272 **4.5.2 Sign Up Page**

1273 First time users will be required to create an account. They are to provide their
1274 name, email and password. For their user type, there are four roles to choose
1275 from: Fisher, Supplier, Retailer and Consumer.

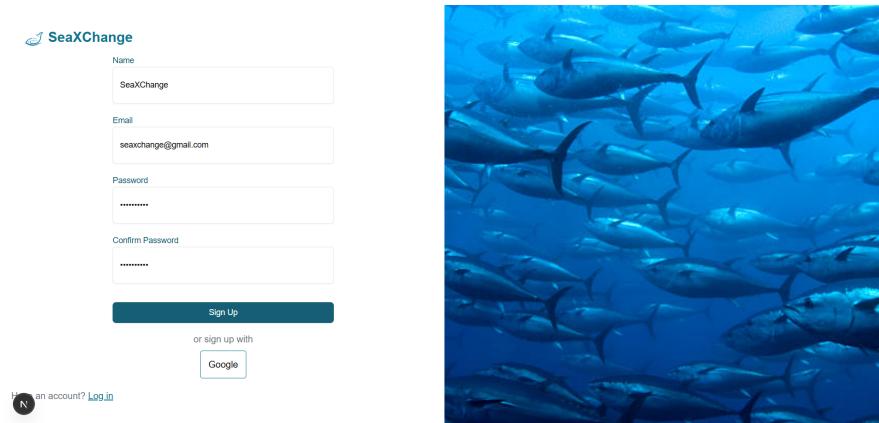


Figure 4.14: SignUp Page

1276 **4.5.3 Login Page**

1277 Returning user are required to login with their email and password.

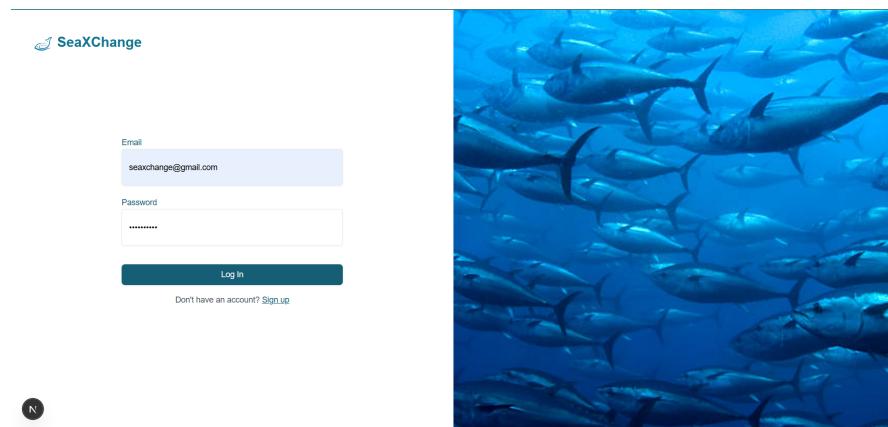


Figure 4.15: LogIn Page

¹²⁷⁸ **4.5.4 Homepage**

¹²⁷⁹ Each user type has their own respective homepages and features.

- ¹²⁸⁰ • **Fisher** Fishers can add a fish catch using the "Add catch" button, where
- ¹²⁸¹ they are to input the species of the fish, weight in kg, catch location, catch
- ¹²⁸² date, fishing method used and vessel name. The remaining text fields like
- ¹²⁸³ the Supplier name, Retailer name and Consumer name are left null and
- ¹²⁸⁴ cannot be edited because they will be filled out by the other users receiving
- ¹²⁸⁵ the tuna asset. Users can send a tuna asset to the Supplier. Users can also
- ¹²⁸⁶ browse existing tuna assets.

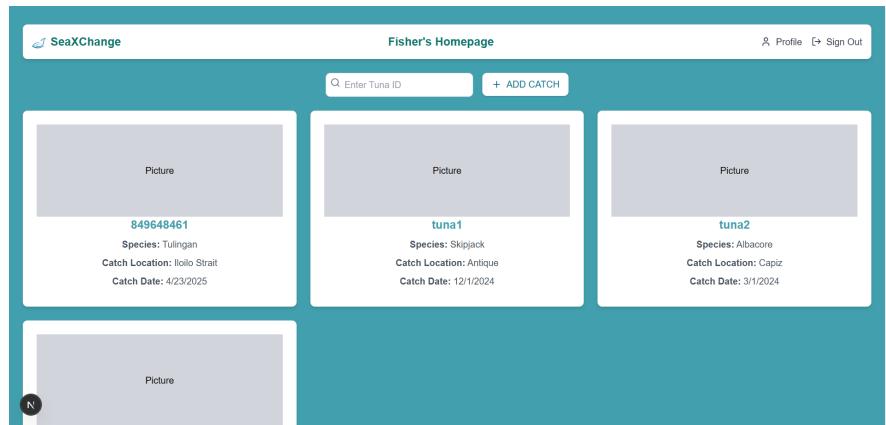


Figure 4.16: Fisher Homepage

- ¹²⁸⁷ • **Supplier** Suppliers can browse existing tuna assets. Upon clicking a tuna
- ¹²⁸⁸ asset, the user can only edit the Supplier text field. They can send the tuna
- ¹²⁸⁹ asset to the Retailer.

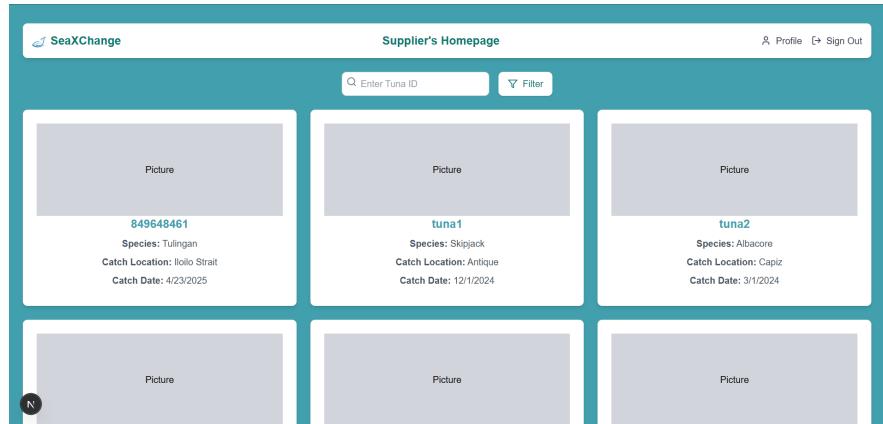


Figure 4.17: Supplier Homepage

- 1290 • **Retailer** Retailers can browse existing tuna assets and can send it to the
1291 Consumer.

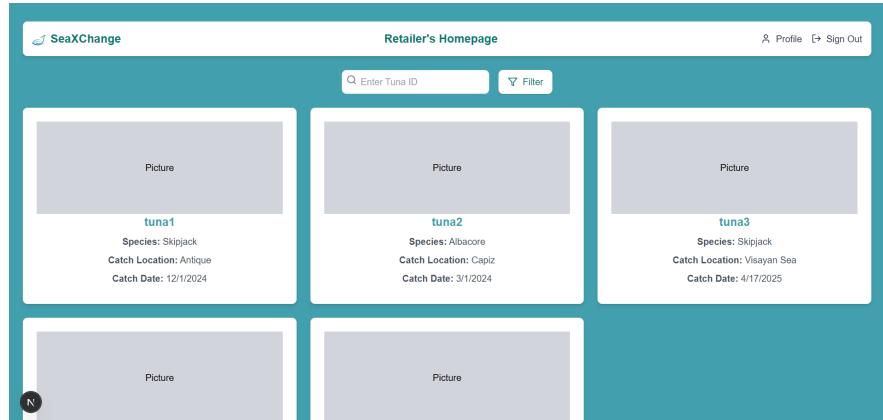


Figure 4.18: Retailer Homepage

- 1292 • **Consumer** Consumers can only view the tuna asset and cannot edit any-
1293 thing else

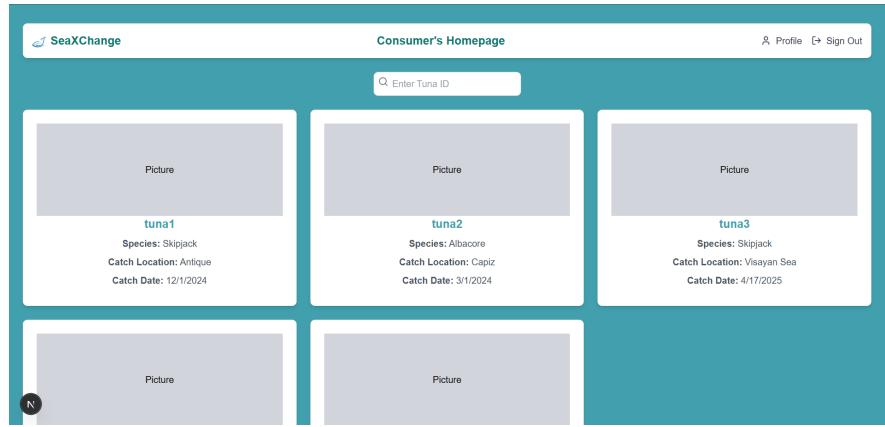


Figure 4.19: Consumer Homepage

1294 4.5.5 Profile

1295 The user's profile information is shown on the homepage through a pop-up. It
1296 shows the user's name and role.

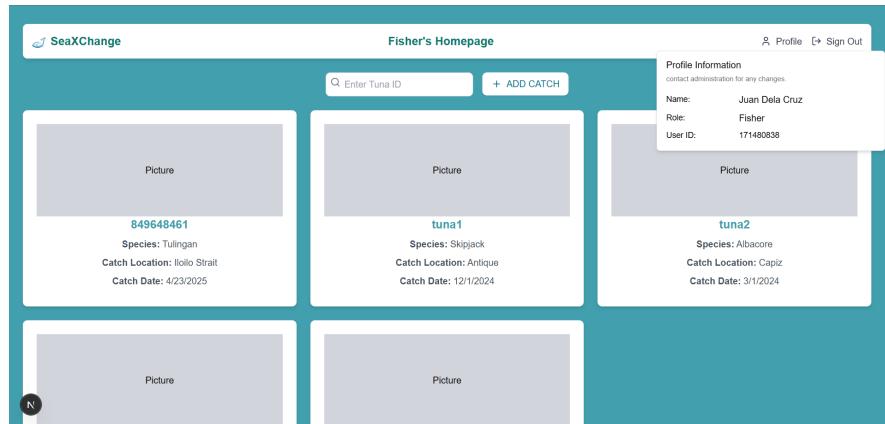


Figure 4.20: View Profile

1297 4.5.6 Logout

1298 Users can logout of their accounts and is redirected to the Signup Page.

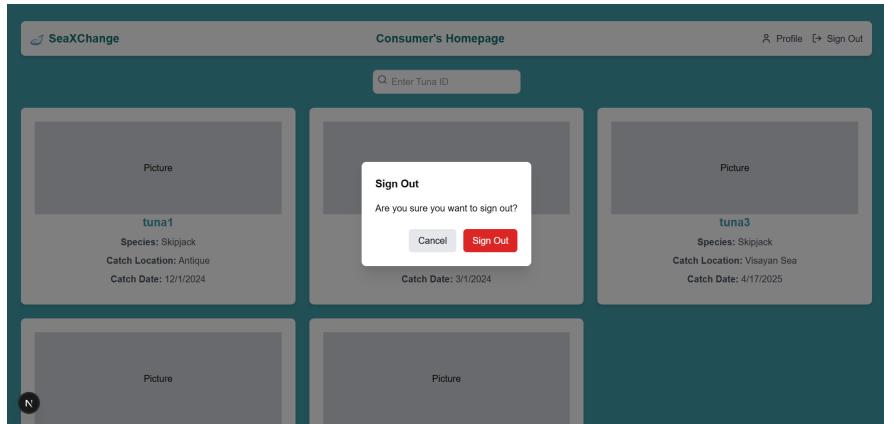


Figure 4.21: Log Out

1299 4.6 System Discussion

1300 After modifying the Hyperledger Fabric smart contract to assess necessary pro-
1301 cesses involved in the tuna supply chain, the blockchain is ready to be invoked
1302 wherein the smart contract can be activated. To start, a new tuna asset is added
1303 and registered to the blockchain. Each tuna asset has its attributes or details.
1304 Before proceeding to the transfer of tuna asset, the smart contract is queried to
1305 verify if the creation of the asset is successful and if it is part of the current in-
1306 ventory. After that, the tuna asset can be transferred from fisher to supplier and
1307 the asset's owner is updated. The smart contract is queried again to verify if the
1308 asset details have been updated successfully. With the same process, the tuna as-
1309 set is transferred from supplier to retailer using the smart contract and the owner
1310 is updated again. To ensure that the asset details are successfully updated, the
1311 smart contract is queried again. The final step is to query the smart contract to
1312 show the overview of all the assets in the supply chain. With this, it can be seen
1313 all the tuna assets from fishing to retail. Overall, the steps and process provides

¹³¹⁴ transparency and traceability in the tuna supply chain.

¹³¹⁵ 4.7 User Demonstration and Feedback Results

¹³¹⁶ 4.7.1 Demo Setup and Scenario

¹³¹⁷ During the demonstration of the system, the participants had a brief introduction
¹³¹⁸ of the key functionalities of the SeaXChange app. They were shown how to
¹³¹⁹ create an account, input and send tuna assets from one stakeholder to another.

¹³²⁰ Participants were also shown how real-time updates were reflected on the app.

¹³²¹ Finally, they were introduced on how to view transaction histories and traceability
¹³²² information on each tuna asset. Throughout the demonstration, participants
¹³²³ were encouraged to ask questions and provide feedback on the usability and func-
¹³²⁴ tionality of the system. After the demonstration, they were given feedback forms
¹³²⁵ in order to assess the SeaXChange app.

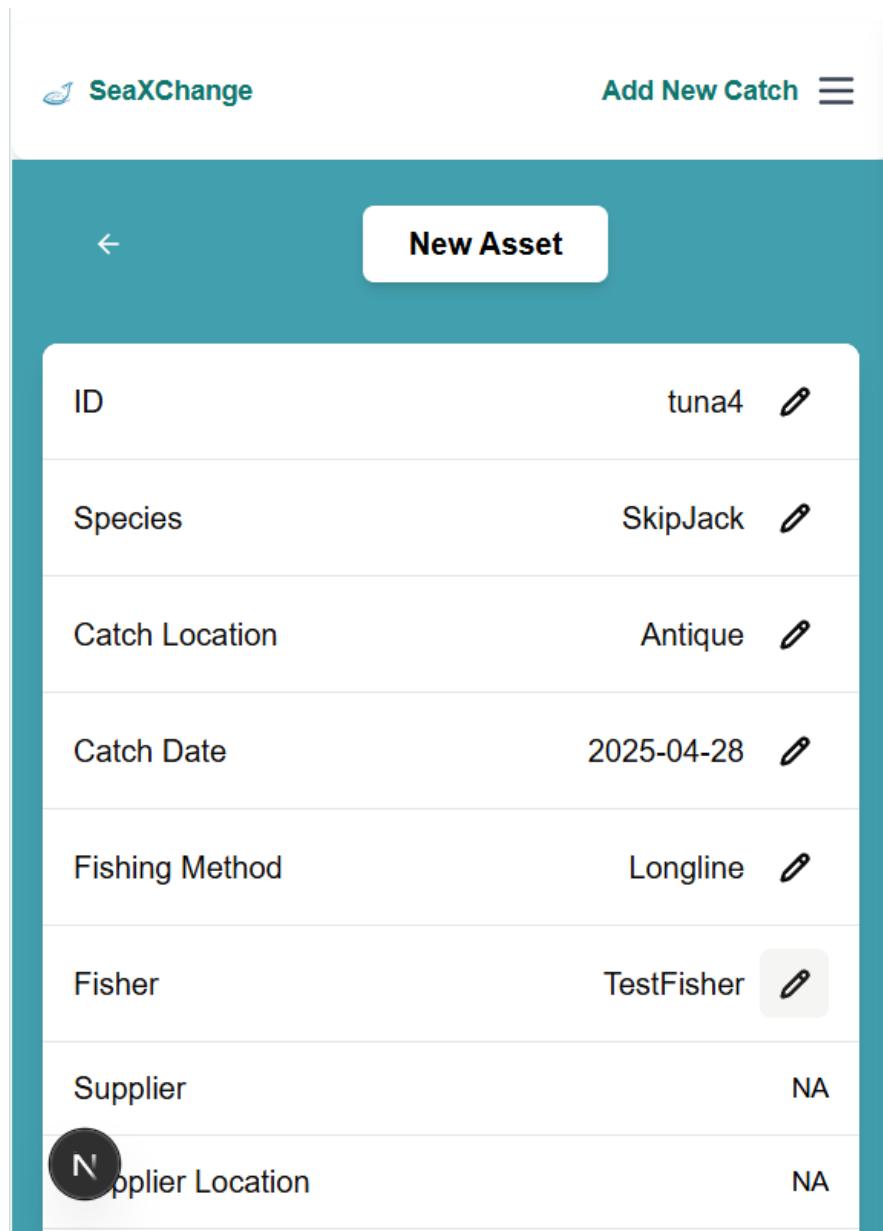


Figure 4.22: Add Catch (Asset)

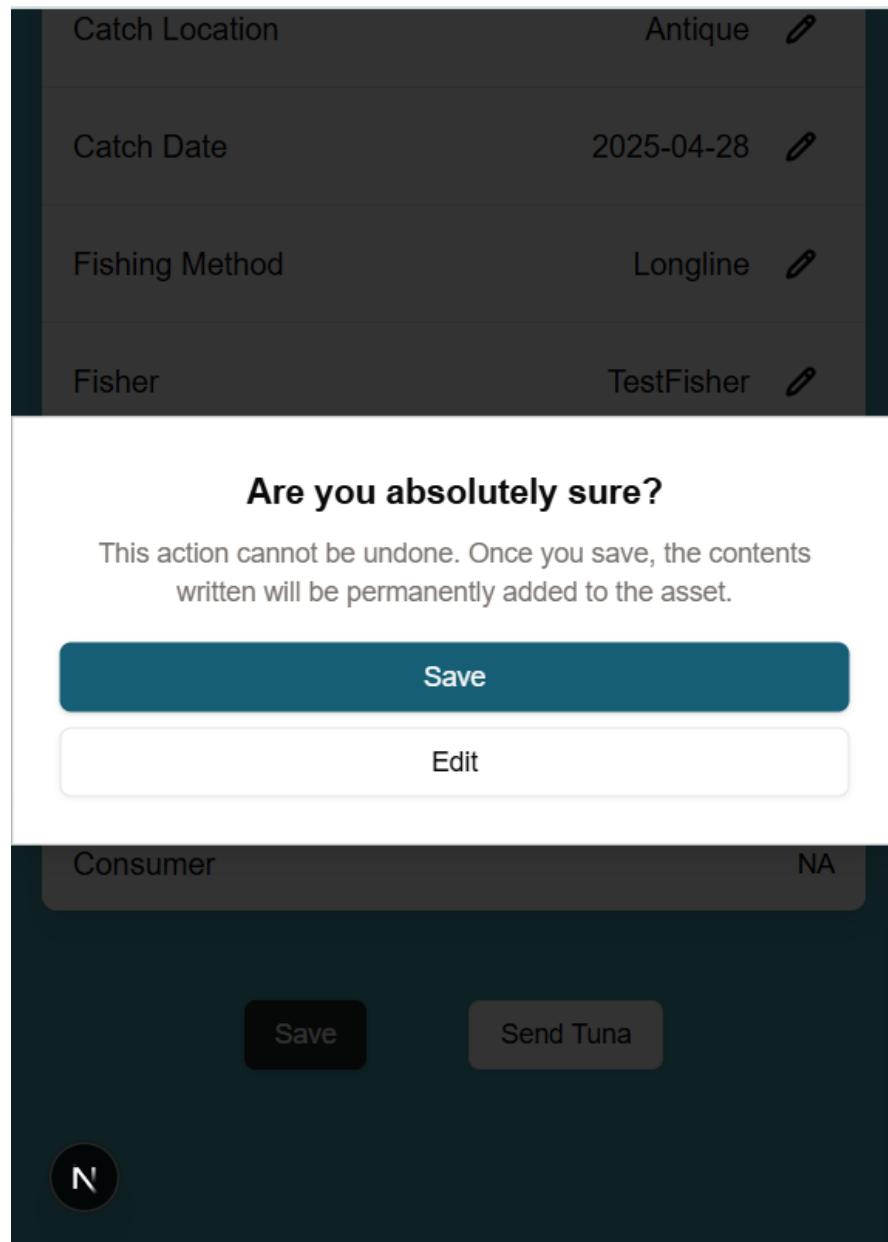


Figure 4.23: Save Details

Species	SkipJack	
Catch Location	Antique	
Catch Date	2025-04-28	
Fishing Method	Longline	
Fisher	TestFisher	
Supplier	NA	
Supplier Location	NA	
Retailer	NA	
Retailer Location	NA	
Consumer	NA	

Send Tuna

Figure 4.24: After Save Details

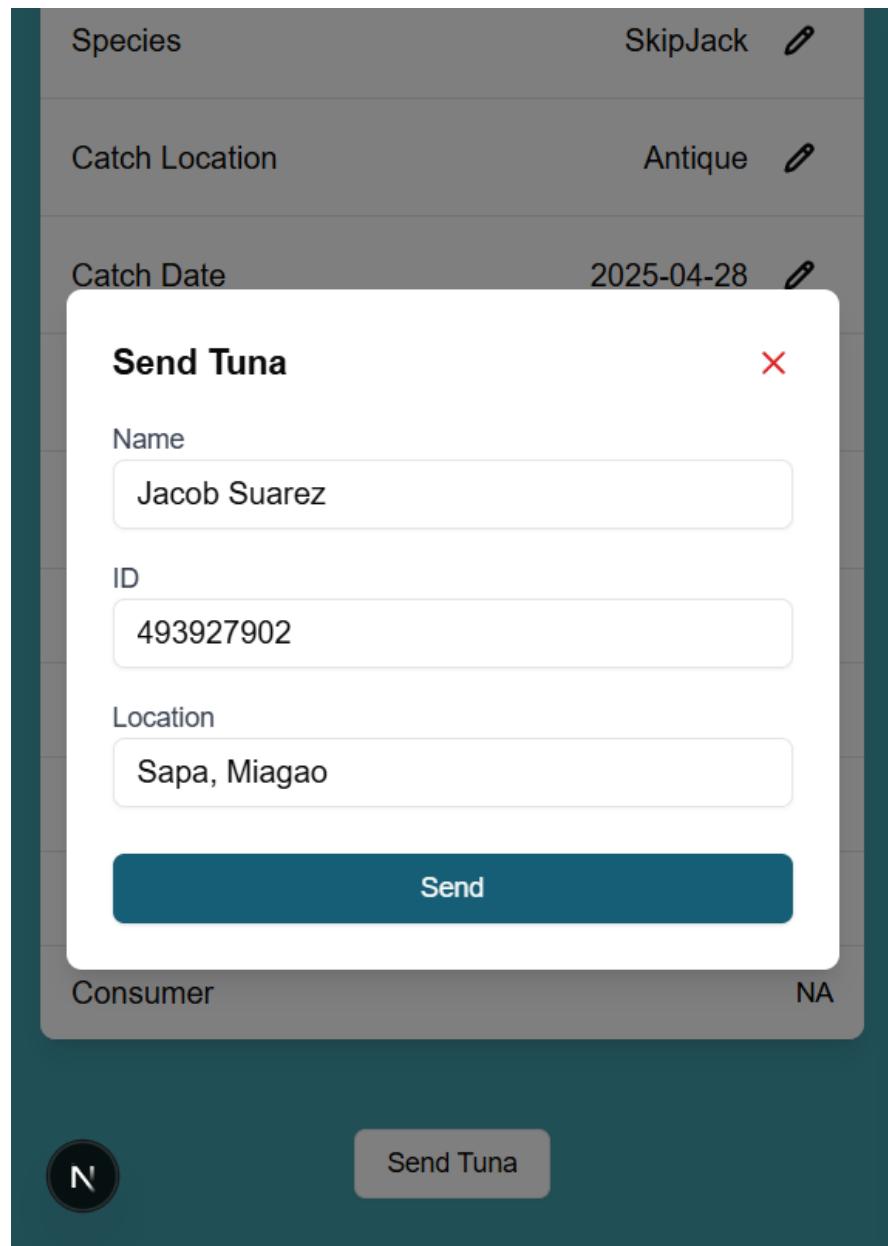


Figure 4.25: Send Asset

¹³²⁶ **4.7.2 Summarized Feedback**

Functionality	Stakeholder	Mean	Description
Track assets	Entire Group	3.67	Average
	Fishermen	4.0	Good
	Supplier and Retailers	3.0	Average
	Consumers	4.0	Good
Verify tuna assets	Entire Group	3.67	Average
	Fishermen	3.33	Average
	Supplier and Retailers	4.0	Good
	Consumers	3.67	Average
Support real-time updates	Entire Group	3.56	Average
	Fishermen	3.78	Average
	Supplier and Retailers	4.0	Good
	Consumers	4.0	Good
Enable smart contract execution	Entire Group	3.42	Average
	Fishermen	2.33	Fair
	Supplier and Retailers	3.25	Average
	Consumers	4.67	Good

Table 4.2: Mean ratings and descriptions for functionality-related features per stakeholder group.

¹³²⁷ When taken as a whole, the respondents have average feedback in asset tracking
¹³²⁸ but when classified by stakeholder, the fishermen ($M = 4.0$) and consumers (M
¹³²⁹ $= 4.0$) had good feedback in tracking , while the supplier and retailers have an
¹³³⁰ average rating ($M = 3.0$). For verifying tuna assets, the entire group has an average
¹³³¹ feedback. When classified by stakeholder, the fishermen ($M = 3.33$) and consumers
¹³³² ($M = 3.67$) have average ratings. For real-time updates, the respondents, when
¹³³³ taken as a whole, have an average feedback. When classified by stakeholder, the
¹³³⁴ fishermen ($M = 3.78$) have an average rating, while both supplier and retailers (M
¹³³⁵ $= 4.0$) and consumers ($M = 4.0$) have good ratings. For smart contract execution,
¹³³⁶ the respondents, when taken as a whole, also have an average feedback. When
¹³³⁷ classified according to stakeholder, the fishermen have a fair rating ($M = 2.33$),

- ¹³³⁸ the supplier and retailers have average ratings ($M = 3.25$) and the consumers have
¹³³⁹ good ratings ($M = 4.67$).

End-user Needs	Stakeholder	Mean	Description
Provide transparency in tracking	Entire Group	3.56	Average
	Fishermen	2.67	Fair
	Supplier and Retailers	4.0	Good
	Consumers	4.0	Good
Provide seamless interaction	Entire Group	3.77	Average
	Fishermen	1.33	Poor
	Supplier and Retailers	3.0	Average
	Consumers	4.0	Good

Table 4.3: Mean ratings and descriptions for end-user needs-related features per stakeholder group.

- ¹³⁴⁰ The respondents, when taken as a whole, had an average feedback in transparency.
¹³⁴¹ When classified by stakeholder, The fishermen have fair ratings ($M = 2.67$), while
¹³⁴² both supplier and retailers ($M = 4.0$) and consumers ($M = 4.0$) have good ratings.
¹³⁴³ In evaluating the seamless interaction of the app, the entire group has an average
¹³⁴⁴ feedback ($M = 3.77$). When classified by stakeholder, the fishermen ($M = 1.33$)
¹³⁴⁵ have poor feedback, the supplier and retailers have average feedback ($M = 3.0$)
¹³⁴⁶ and the consumers have good feedback ($M = 4.0$) in seamless interaction.

Performance	Stakeholder	Mean	Description
Processes transactions efficiently	Entire Group	3.81	Average
	Fishermen	3.67	Average
	Supplier and Retailers	3.75	Average
	Consumers	4.0	Good
Ensures data integrity and security	Entire Group	3.31	Average
	Fishermen	2.67	Fair
	Supplier and Retailers	3.25	Average
	Consumers	4.0	Good

Table 4.4: Mean ratings and descriptions for performance-related features per stakeholder group.

1347 As a whole, the respondents have an average feedback on efficient transactions (M
 1348 = 3.81). If evaluated per stakeholder, both fishermen ($M = 3.67$), supplier and
 1349 retailers ($M = 3.75$) evaluated average while consumers had good feedback ($M =$
 1350 4.0). For data security, the entire group has an average feedback ($M = 3.31$). The
 1351 fishermen have fair evaluation ($M = 2.67$), supplier and retailers ($M = 3.25$) have
 1352 an average and consumers have solid scores ($M = 4.0$).

Usability	Stakeholder	Mean	Description
Provides intuitive interface	Entire Group	3.83	Average
	Fishermen	4.0	Good
	Supplier and Retailers	3.5	Average
	Consumers	4.0	Good
Allows cross-platform accessibility	Entire Group	4.14	Good
	Fishermen	4.0	Good
	Supplier and Retailers	3.75	Average
	Consumers	4.67	Good
Clear, structured, and visually appealing info	Entire Group	3.80	Average
	Fishermen	3.33	Average
	Supplier and Retailers	3.75	Average
	Consumers	4.33	Good

Table 4.5: Mean ratings and descriptions for usability-related features per stakeholder group.

1353 It shows the frequency of intuitive interface among the respondents when taken
 1354 as a whole is average ($M = 3.83$). When classified according to stakeholder,
 1355 both fishermen ($M = 4.0$) and consumers ($M = 4.0$) have good ratings, while the
 1356 supplier and retailers ($M = 3.5$) have average ratings. For cross-platform usage,
 1357 the entire group rated good ($M = 4.14$). When classified according to stakeholder,
 1358 both fishermen ($M = 4.0$) and consumers ($M = 4.1$) also have good ratings, while
 1359 supplier and retailers ($M = 3.75$) have average. For visual clarity, the entire group
 1360 rated average ($M = 3.80$). When classified by each stakeholder, both fishermen
 1361 ($M = 3.33$) and supplier and retailers ($M = 3.75$) have average ratings, while

₁₃₆₂ consumers ($M = 4.33$) have good ratings.

Ease of Use	Stakeholder	Mean	Description
Clear instructions for new users	Entire Group	3.89	Average
	Fishermen	4.0	Good
	Supplier and Retailers	4.0	Good
	Consumers	3.67	Average
Uses clear and simple language	Entire Group	3.31	Average
	Fishermen	4.03	Good
	Supplier and Retailers	3.75	Average
	Consumers	4.33	Good

Table 4.6: Mean ratings and descriptions for ease of use-related features per stakeholder group.

1363 When taken as a whole, the respondents ($M = 3.89$) rated instruction clarity as
 1364 average. When classified by stakeholder, both fishermen ($M = 4.0$) and supplier
 1365 and retailers ($M = 4.0$) have good feedback regarding instruction clarity, while the
 1366 consumers ($M = 3.67$) have average feedback. The entire group rated language
 1367 clarity as average ($M = 3.31$). When evaluated by each stakeholder, both fisher-
 1368 men ($M = 4.03$) and consumers ($M = 4.33$) have good feedback, while supplier
 1369 and retailers ($M = 3.75$) have average feedback.

Feasibility	Stakeholder	Mean	Description
Integration with tuna industry	Entire Group	4.06	Good
	Fishermen	4.0	Good
	Supplier and Retailers	4.5	Good
	Consumers	3.67	Average
Consumer use to track tuna products	Entire Group	4.03	Good
	Fishermen	4.0	Good
	Supplier and Retailers	3.75	Average
	Consumers	4.33	Good

Table 4.7: Mean ratings and descriptions for feasibility-related features per stakeholder group.

1370 When taken as a whole, it shows that the respondents have good feedback in the
 1371 system integration. When classified by each stakeholder, both the fishermen ($M =$

1372 4.0) and supplier and retailers ($M = 4.5$) have good feedback in system integration,
1373 while the consumers($M = 3.67$) have an average rating. The frequency of consumer
1374 use among stakeholders, when taken as a whole, have good feedback ($M = 4.03$).
1375 When analyzed individually, both the fishermen ($M = 4.0$) and consumers ($M =$
1376 4.33) have a good rating, while an average rating for the supplier and retailers (M
1377 = 3.75).

1378 4.7.3 Results and Analysis

1379 For functionality, fishermen rated asset tracking positively but identified chal-
1380 lenges in real-time updates and smart contract execution. Suppliers and retail-
1381 ers similarly found strengths in verification and real-time updates, although as-
1382 set tracking averaged slightly lower. Consumers consistently rated functionality
1383 highly, particularly appreciating the system's ability to support smart contracts.

1384 Regarding end-user needs, fishermen provided the lowest ratings, highlighting
1385 difficulties with seamless interaction, considering the digital barrier that they ex-
1386 perience while interacting with the app. Suppliers, retailers, and consumers gave
1387 higher scores, suggesting that while transparency was generally achieved, the sys-
1388 tem's ease of interaction required further improvement for all users.

1389 In terms of performance, fishermen and suppliers reported average transaction
1390 efficiency but noted concerns regarding data integrity and security. Consumers,
1391 on the other hand, expressed confidence in both areas.

1392 For usability, consumers provided the highest ratings, emphasizing the intuitive
1393 design and accessibility of the interface. Fishermen and suppliers also rated us-

¹³⁹⁴ ability positively but suggested improvements, such as better visual appeal and
¹³⁹⁵ language localization. Feedbacks suggested incorporating the Karay-a language
¹³⁹⁶ since most of the potential users uses that language in their everyday lives. An-
¹³⁹⁷ other suggestion was to use capitalization (UI/UX) for the name of the tuna to
¹³⁹⁸ give more emphasis to it.

¹³⁹⁹ Ease of use was generally rated positively across all groups. However, feedback
¹⁴⁰⁰ highlighted the need for additional user guidance features, including clearer button
¹⁴⁰¹ descriptions and localized instructions for broader accessibility.

¹⁴⁰² Finally, feasibility was strongly affirmed by all groups. Stakeholders believed the
¹⁴⁰³ system could be effectively integrated into the existing tuna supply chain and
¹⁴⁰⁴ accessed by consumers as a means to monitor tuna products.

¹⁴⁰⁵ Chapter 5

¹⁴⁰⁶ Conclusion

¹⁴⁰⁷ 5.1 Overview

¹⁴⁰⁸ This chapter presents the key findings and conclusion of the study. It also discusses
¹⁴⁰⁹ how the system addresses the challenges of traceability within the tuna supply
¹⁴¹⁰ chain. This chapter also provides recommendations in enhancing the system's
¹⁴¹¹ functionality and usability, ensuring that the SeaXChange app continues to meet
¹⁴¹² evolving needs of its users.

¹⁴¹³ 5.2 Conclusion

¹⁴¹⁴ This study aimed to develop and evaluate SeaXChange, which is a blockchain-
¹⁴¹⁵ driven app designed to enhance transparency, traceability and accountability
¹⁴¹⁶ within the tuna supply chain. Through the adaption of Scrum, the team was

¹⁴¹⁷ able to develop a functional prototype that was based on iterative development
¹⁴¹⁸ to achieve goals.

¹⁴¹⁹ The results from the gathered data suggests that the app has effectively addressed
¹⁴²⁰ key challenges in traceability and accountability, especially among suppliers and
¹⁴²¹ consumers. The stakeholders consistently rated the system as good. However,
¹⁴²² some areas need improvement, especially in ensuring seamless interaction and
¹⁴²³ data security for fishermen.

¹⁴²⁴ Overall, SeaXChange demonstrates strong potential as a technological solution for
¹⁴²⁵ promoting transparency, combating illegal, unreported, and unregulated (IUU)
¹⁴²⁶ fishing, and empowering stakeholders across the tuna supply chain.

¹⁴²⁷ 5.3 Recommendations

¹⁴²⁸ After analyzing and interpreting the gathered data, the researchers had identified
¹⁴²⁹ some improvements and recommendations for the further development and
¹⁴³⁰ implementation of the SeaXChange app.

¹⁴³¹ 1. Incorporation of Local Language

¹⁴³² Since most of the target users are within Miagao, the system could provide
¹⁴³³ multi-language support, including the Karay-a language. This will improve
¹⁴³⁴ guidance through clearer button descriptions.

¹⁴³⁵ 2. Utilization of IoT

¹⁴³⁶ The system could use Internet of Things (IoT) in verifying the fisherman's
¹⁴³⁷ location. This will add more accountability in tracing the fisherman's cur-

1438 rent location.

1439 **3. Inclusion of User Manual**

1440 To further enhance the experience of its users, the researcher could provide
1441 printed or digital copies on the system's functionalities. This will help users
1442 navigate through the system without being lost.

1443 **5.4 Summary**

1444 In conclusion, the development of the SeaXChange app highlights the critical role
1445 of emerging technologies in providing solution regarding the traceability, trans-
1446 parency and accountability within the tuna supply chain. While the system has
1447 demonstrated strong potential, continuous improvements are still needed to en-
1448 sure its effectiveness. Moreover, further development and usability enhancements
1449 will be essential in attaining SeaXChange's goal of creating a more traceable,
1450 transparent and accountable tuna industry.

¹⁴⁵¹

Chapter 6

¹⁴⁵²

References

- ¹⁴⁵³ Ahamed, N. N., Karthikeyan, P., Anandaraj, S., & Vignesh, R. (2020). *Sea food supply chain management using blockchain.* <https://ieeexplore.ieee.org/abstract/document/9074473>.
- ¹⁴⁵⁴
- ¹⁴⁵⁵
- ¹⁴⁵⁶ Ali, V., Norman, A. A., & Azzuhri, S. R. B. (2023). *Characteristics of blockchain and its relationship with trust.* <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10041154>.
- ¹⁴⁵⁷
- ¹⁴⁵⁸
- ¹⁴⁵⁹ Anitha, R., & Sankarasubramanian, R. (2006). Verifiable encryption of digital signatures using elliptic curve digital signature algorithm and its implementation issues. In *Innovations in information technology: Research and application.* doi: 10.4018/978-1-59904-168-1.ch010
- ¹⁴⁶⁰
- ¹⁴⁶¹
- ¹⁴⁶²
- ¹⁴⁶³ Aponte, M., Anastasio, A., Marrone, R., Mercogliano, R., Peruzy, M. F., & Murru, N. (2018). *Impact of gaseous ozone coupled to passive refrigeration system to maximize shelf-life and quality of four different fresh fish products.* <https://www.sciencedirect.com/science/article/pii/S0023643818302974>.
- ¹⁴⁶⁴
- ¹⁴⁶⁵
- ¹⁴⁶⁶
- ¹⁴⁶⁷ Asche, F., Garlock, T. M., Anderson, J. L., Bush, S. R., Smith, M. D., Anderson,

- 1468 C. M., ... Vannuccini, S. (2018). *Three pillars of sustainability in fisheries*.
1469 <https://www.pnas.org/doi/abs/10.1073/pnas.1807677115>.
- 1470 Cocco, L., & Mannaro, K. (2021). *Blockchain in agri-food traceability systems: a model proposal for a typical italian food product*. <https://ieeexplore.ieee.org/abstract/document/9425984>.
- 1471
- 1472
- 1473 Cole, R., Stevenson, M., & Aitken, J. (2019). *Blockchain technology: implications for operations and supply chain management*. <https://www.emerald.com/insight/content/doi/10.1108/SCM-09-2018-0309/full/html#sec006>.
- 1474
- 1475
- 1476 Cordova, R. S., Maata, R. L. R., Epoc, F. J., & Alshar'e, M. (2021). *Challenges and opportunities of using blockchain in supply chain management*. <http://www.gbmrrjournal.com/pdf/v13n3/V13N3-18.pdf>.
- 1477
- 1478
- 1479 CSCMP. (2024). *Cscmp supply chain management definitions and glossary*. https://cscmp.org/CSCMP/CSCMP/Educate/SCM_Definitions_and_Glossary_of_Terms.aspx.
- 1480
- 1481
- 1482 Delfino, A. N. (2023). *Analysing the value chain of skipjack tuna(katsuwonus pelamis) in partido district, camarinessur, philippines*. <https://www.inderscienceonline.com/doi/epdf/10.1504/IJVCM.2023.129271>.
- 1483
- 1484
- 1485 Digal, L. N., Placencia, S. G. P., & Balgos, C. Q. (2017). *Market assessment on the incentives and disincentives for the adoption of sustainable practices along the tuna value chain in region 12, philippines*. <https://www.sciencedirect.com/science/article/abs/pii/S0308597X17301197>.
- 1486
- 1487
- 1488
- 1489 Grantham, A., Pandan, M. R., Roxas, S., & Hitchcock, B. (2022). *Overcoming catch data collection challenges and traceability implementation barriers in a sustainable, small-scale fishery*. <https://www.mdpi.com/2071-1050/14/3/1179>.
- 1490
- 1491
- 1492
- 1493 Hackius, N., & Petersen, M. (2017). *Blockchain in logistics and supply chain: Trick*

- 1494 or treat? <https://www.researchgate.net/publication/318724655>
 1495 _Blockchain_in_Logistics_and_Supply_Chain_Trick_or_Treat.
- 1496 Hugos, M. H. (2024). *Essentials of supply chain management* (Edition Number (5) ed.). Hoboken, New Jersey: Wiley. Retrieved from https://books.google.com.ph/books?hl=en&lr=&id=zpz0EAAAQBAJ&oi=fnd&pg=PP7&dq=supply+chain+&ots=jAuHDxF99j&sig=z10Tue18LKtl3pIQWcr2uZT4pRw&redir_esc=y#v=onepage&q=supply%20chain&f=false
- 1501 Hyperledger Foundation. (2024). Hyperledger fabric documentation [Computer software manual]. Retrieved from <https://hyperledger-fabric.readthedocs.io/> (Available at <https://hyperledger-fabric.readthedocs.io/>)
- 1506 Islam, S., & Cullen, J. M. (2021). *Food traceability: A generic theoretical framework*. <https://doi.org/10.1016/j.foodcont.2020.107848>.
- 1508 Khan, S. N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E., & Bani-Hani, A. (2021). *Blockchain smart contracts: Applications, challenges, and future trends*. <https://doi.org/10.1007/s12083-021-01127-0>.
- 1511 Kresna, B. A., Seminar, K. B., & Marimin, M. (2017). *Developing a traceability system for tuna supply chains*. <http://ijis-scm.bsne.ch/ojs.excelingtech.co.uk/index.php/IJSCM/article/view/1584/0.html>.
- 1514 Larissa, S., & Parung, J. (2021). *Designing supply chain models with blockchain technology in the fishing industry in indonesia*. <https://iopscience.iop.org/article/10.1088/1757-899X/1072/1/012020>.
- 1517 Llanto, G. M., Ortiz, M. K. P., & Madriaga, C. A. D. (2018). *The philippines' tuna industry*. https://www.eria.org/uploads/media/RURB_2018_FullReport.pdf#page=221.

- 1520 Macusi, E. D., Castro, M. M. C., Nallos, I. M., & Perales, C. P. (2023).
1521 *Fishers' communication as a critical factor for tuna catches and potential*
1522 *benefits of traceability draws small-scale fishers to program.* <https://www.sciencedirect.com/science/article/pii/S0964569123003873>.
- 1523
- 1524 Macusi, E. D., da Costa-Neves, A. C., Tipudan, C. D., & Babaran, R. P. (2023).
1525 *Closed season and the distribution of small-scale fisheries fishing effort in*
1526 *davao gulf, philippines.* <https://www.mdpi.com/2673-4060/4/1/4>.
- 1527 Mercogliano, R., & Santonicola, S. (2019). *Scombroid fish poisoning: Factors influencing the production of histamine in tuna supply chain. a review.* <https://www.sciencedirect.com/science/article/pii/S0023643819307169>.
- 1528
- 1529
- 1530
- 1531 Mullon, C., Guillotreau, P., Galbraith, E. D., Fortilus, J., Chaboud, C., Bopp, L.,
1532 ... Kaplan, D. M. (2017). *Exploring future scenarios for the global supply*
1533 *chain of tuna.* <https://doi.org/10.1016/j.dsr2.2016.08.004>.
- 1534 Nassar, M., Rottenstreich, O., & Orda, A. (2024, February). Cfto:
1535 Communication-aware fairness in blockchain transaction ordering. *IEEE*
1536 *Transactions on Network and Service Management*, 21(1), 490–506. doi:
1537 10.1109/TNSM.2023.3298201
- 1538 Nepomuceno, L. T., Bacordo, R. S., Camu, D. G. Y., & Ramiscal, R. V. (2020).
1539 *Abundance, distribution, and diversity of tuna larvae (family scombridae)*
1540 *in the philippine waters.* [https://www.nfrdi.da.gov.ph/tpjf/vol27_2/Abundance,%20Distribution,%20and%20Diversity%20of%20Tuna%20Larvae%20\(Family%20Scombridae\)%20in%20the%20Philippine%20waters.pdf](https://www.nfrdi.da.gov.ph/tpjf/vol27_2/Abundance,%20Distribution,%20and%20Diversity%20of%20Tuna%20Larvae%20(Family%20Scombridae)%20in%20the%20Philippine%20waters.pdf).
- 1541
- 1542
- 1543
- 1544 Obura, D. O., Aeby, G., Amornthammarong, N., Appeltans, W., Bax,
1545 N., Bishop, J., ... Wongbusarakum, S. (2019). *Coral reef*

- 1546 *monitoring, reef assessment technologies, and ecosystem-based man-*
 1547 *agement.* [https://www.frontiersin.org/journals/marine-science/](https://www.frontiersin.org/journals/marine-science/articles/10.3389/fmars.2019.00580/full)
 1548 *articles/10.3389/fmars.2019.00580/full.*
- 1549 Pacoma, A. U., & Yap-Dejeto, L. G. (2019). *Health risk assessment: Total
 1550 mercury in canned tuna and in yellowfin and frigate tuna caught from leyte
 1551 gulf and philippine sea.* https://www.researchgate.net/publication/340827869_Health_Risk_Assessment_Total_Mercury_in_Canned_Tuna_and_in_Yellowfin_and_Frigate_Tuna_Caught_from_Leyte_Gulf_and_Philippine_Sea.
- 1552 Paillin, D., Tupan, J., Paillin, J., Latuny, W., & Lawalata, V. (2022).
 1553 *Risk assessment and risk mitigation in a sustainable tuna sup-*
 1554 *ply chain.* https://www.actalogistica.eu/issues/2022/I_2022_06_Paillin_Tupan_Paillin_Latuny_Lawalata.pdf.
- 1555 Parenrenge, S. M., Pujawan, N., Karningsih, P. D., & Engelseth, P. (2016).
 1556 *Mitigating risk in the tuna supply through traceability system develop-*
 1557 *ment.* https://himolde.brage.unit.no/himolde-xmlui/bitstream/handle/11250/3052893/Cristin-id_1318891_engelseth.pdf?sequence=1&isAllowed=y.
- 1558 Peruzy, M. F., Murru, N., Perugini, A. G., Capuano, F., Delibato, E., Mer-
 1559 cogliano, R., ... Proroga, Y. T. R. (2017). *Evaluation of virulence genes
 1560 in yersinia enterocolitica strains using sybr green real-time pcr.* <https://www.sciencedirect.com/science/article/pii/S0740002016304555>.
- 1561 Pomeroy, R., & Courtney, C. A. (2018). *The philippines context for marine
 1562 tenure and small-scale fisheries.* <https://sci-hub.st/https://doi.org/10.1016/j.marpol.2018.05.030>.
- 1563 Sarmah, S. S. (2018). *Understanding blockchain technology.* <http://article>

- 1572 .sapub.org/10.5923.j.computer.20180802.02.html.
- 1573 1574 1575 Shamsuzzoha, A., Marttila, J., & Helo, P. (2023). *Blockchain-enabled traceability system for the sustainable seafood industry*. <https://www.tandfonline.com/doi/full/10.1080/09537325.2023.2233632#d1e340>.
- 1576 1577 1578 1579 Smithrithee, M., Sato, A., Wanchana, W., Tongdee, N., Sulit, V. T., & Saraphaivanich, K. (2020). *Pushing for the elimination of iuu fishing in the southeast asian region*. <http://repository.seafdec.org/handle/20.500.12066/6610>.
- 1580 1581 1582 1583 1584 Snyder, L. V., & Shen, Z.-J. M. (2019). *Fundamentals of supply chain theory* (Edition Number (2) ed.). New York: Wiley. Retrieved from https://books.google.com.ph/books?hl=en&lr=&id=sJSaDwAAQBAJ&oi=fnd&pg=PR21&dq=supply+chain+&ots=IDNDcy0t37&sig=ssnh-6IDRAi1JzBRpohxT-hiwTE&redir_esc=y#v=onepage&q=supply
- 1585 1586 Tiwari, U. (2020). *Application of blockchain in agri-food supply chain*. <https://doi.org/10.33258/bioex.v2i2.233>.
- 1587 1588 1589 1590 Verkhivker, Y., & Altmann, E. (2018). *Influence parameters of storage on process of formation the histamine in fish and fish products*. <https://www.sciencepublishinggroup.com/article/10.11648/j.wros.20180701.12>.
- 1591 1592 1593 1594 Xu, J., Wang, W., Zeng, Y., Yan, Z., & Li, H. (2022). Raft-plus: Improving raft by multi-policy based leader election with unprejudiced sorting. *Symmetry*, 14(6), 1122. Retrieved from <https://doi.org/10.3390/sym14061122> doi: 10.3390/sym14061122
- 1595 1596 1597 Yang, Y.-C., & Lin, H.-Y. (2017). *Cold supply chain of longline tuna and transport choice*. <https://www.emerald.com/insight/content/doi/10.1108/mabr-11-2017-0027/full/html>.

1598 Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). *An overview of*
1599 *blockchain technology: Architecture, consensus, and future trends.* [https://
1600 ieeexplore.ieee.org/abstract/document/8029379.](https://ieeexplore.ieee.org/abstract/document/8029379)

1601 **Appendix A**

1602 **Code Snippets**

```
1603
1604     const checkAssetAccess = (
1605         asset, userIdentifier, role
1606     ) => {
1607
1608         switch (role.toLowerCase()) {
1609             case 'fisher':
1610                 if (asset.Fisher === userIdentifier)
1611                     return { hasAccess: true, accessType
1612                         : 'full' };
1613
1614             case 'supplier':
1615                 if (asset.Supplier ===
1616                     userIdentifier)
1617                     return { hasAccess: true, accessType
1618                         : 'full' };
1619
1620             case 'retailer':
```

```

1620:         if (asset.Retailers?.includes(
1621:             userIdentifier))
1622:             return { hasAccess: true, accessType
1623:                 : 'full' };
1624:             break;
1625:         case 'consumer':
1626:             if (asset.Consumers?.includes(
1627:                 userIdentifier))
1628:                 return { hasAccess: true, accessType
1629:                     : 'full' };
1630:             break;
1631:         }
1632:
1633:         if (role.toLowerCase() === 'consumer') {
1634:             return { hasAccess: true, accessType
1635:                 : 'readonly' };
1636:         }
1637:
1638:         return { hasAccess: false, accessType: 'none'
1639:                 };
1640:
1641:     };

```

Listing A.1: Role-based access logic for tuna asset visibility

```

1642
1643:     const grpc = require('@grpc/grpc-js');
1644:     const { connect, hash, signers } = require(
1645:         '@hyperledger/fabric-gateway');
1646:     const crypto = require('node:crypto');

```

```

1647:     const fs = require('node:fs/promises');
1648:     const path = require('node:path');
1649:     const { TextDecoder } = require('node:util');
1650:
1651:     const channelName = envOrDefault('CHANNEL_NAME', 'mychannel');
1652:
1653:     const chaincodeName = envOrDefault('CHAINCODE_NAME', 'basic');
1654:
1655:     const mspId = envOrDefault('MSP_ID', 'Org1MSP');
1656:
1657:     // Path to crypto materials.
1658:     const cryptoPath = envOrDefault(
1659:       'CRYPTO_PATH',
1660:       path.resolve(
1661:         __dirname,
1662:         '..',
1663:         '..',
1664:         '..',
1665:         'test-network',
1666:         'organizations',
1667:         'peerOrganizations',
1668:         'org1.example.com'
1669:       )
1670:     );
1671:
1672:     const keyDirectoryPath = envOrDefault(
1673:       'KEY_DIRECTORY_PATH',

```

```
1674)     path.resolve(
1675)       cryptoPath,
1676)       'users',
1677)       'User1@org1.example.com',
1678)       'msp',
1679)       'keystore'
1680)     )
1681);
1682;
1683)   const certDirectoryPath = envOrDefault(
1684)     'CERT_DIRECTORY_PATH',
1685)     path.resolve(
1686)       cryptoPath,
1687)       'users',
1688)       'User1@org1.example.com',
1689)       'msp',
1690)       'signcerts'
1691)     )
1692);
1693;
1694)   const tlsCertPath = envOrDefault(
1695)     'TLS_CERT_PATH',
1696)     path.resolve(cryptoPath, 'peers', 'peer0.org1.
1697)       example.com', 'tls', 'ca.crt')
1698);
1699;
1700)   const peerEndpoint = envOrDefault('PEER_ENDPOINT', '
```

```

1701           localhost:7051');
1702
1703   const peerHostAlias = envOrDefault('PEER_HOST_ALIAS',
1704                                         , 'peer0.org1.example.com');
1705
1706   const utf8Decoder = new TextDecoder();
1707
1708   const assetId = asset${String(Date.now())};

```

Listing A.2: Node.js configuration for Hyperledger Fabric Gateway

```

1708
1709   type Asset struct {
1710
1711     ID
1712     string `json:"ID"`
1713
1714     Species
1715     string `json:"Species"`
1716
1717     Weight
1718     float64 `json:"Weight"`
1719
1720     CatchLocation
1721     string `json:"CatchLocation"`
1722
1723     CatchDate
1724     string `json:"CatchDate"`
1725
1726     FishingMethod
1727     string `json:"FishingMethod"`
1728
1729     Fisher
1730     string `json:"Fisher"`
1731
1732     Supplier
1733     string `json:"Supplier"`
1734
1735     SellingLocationSupplier
1736     string `json:"SellingLocationSupplier"`

```

```

1728:         Retailers
1729:             [] string `json:"Retailers"
1730:             "
1731:         SellingLocationRetailers []
1732:             string `json:"SellingLocationRetailers"
1733:         Consumers
1734:             [] string `json:"Consumers"
1735:             "
1736:     }
1737:
```

Listing A.3: Asset Data Structure

```

1738
1739: func (s *SmartContract) CreateAsset(ctx contractapi.
1740:                                     TransactionContextInterface, id string, species
1741:                                     string, weight float64, catchLocation string,
1742:                                     catchDate string, fishingMethod string, fisher
1743:                                     string) error {
1744:
1745:     exists, err := s.AssetExists(ctx, id)
1746:
1747:     if err != nil {
1748:
1749:         return err
1750:
1751:     }
1752:
1753:     if exists {
1754:
1755:         return fmt.Errorf("the asset %s
1756:                           already exists", id)
1757:
1758:     }
1759:
1760:     asset := Asset{
1761:
1762:         ID:           id,
1763:         Species:      species,
1764:         Weight:       weight,
1765:         CatchLocation: catchLocation,
1766:         CatchDate:    catchDate,
1767:         FishingMethod: fishingMethod,
1768:         Fisher:       fisher,
1769:     }
1770:
1771:     err = s.createAsset(ctx, asset)
1772:
1773:     if err != nil {
1774:
1775:         return err
1776:
1777:     }
1778:
1779:     return nil
1780:
```

```

1755:             Species:           species,
1756:             Weight:            weight,
1757:             CatchLocation:    catchLocation,
1758:             CatchDate:         catchDate,
1759:             FishingMethod:    fishingMethod,
1760:             Fisher:            fisher,
1761:             Retailers:          [] string{},
1762:             SellingLocationRetailers: []
1763:                           string{},
1764:             Consumers:          [] string{},
1765:         }
1766:         assetJSON, err := json.Marshal(asset)
1767:         if err != nil {
1768:             return err
1769:         }
1770:
1771:         return ctx.GetStub().PutState(id, assetJSON)
1772:
1773}

```

Listing A.4: CreateAsset Function

```

1774
1775:     func (s *SmartContract) TransferAsset(ctx
1776:                                         contractapi.TransactionContextInterface, id
1777:                                         string, role string, newParticipant string,
1778:                                         newLocation string) (string, error) {
1779:
1780:         asset, err := s.ReadAsset(ctx, id)
1781:
1782:         if err != nil {
1783:
1784:             return "", fmt.Errorf("failed to

```

```
1782             fetch asset: %v", err)
1783         }
1784
1785     switch role{
1786         case "Supplier":
1787             oldSupplier := asset.Supplier
1788             asset.Supplier = newParticipant
1789             asset.SellingLocationSupplier =
1790                 newLocation
1791             return oldSupplier, s.SaveAsset(ctx,
1792                     id, asset)
1793         case "Retailer":
1794             if !contains(asset.Retailers,
1795                         newParticipant){
1796                 asset.Retailers = append(
1797                     asset.Retailers,
1798                     newParticipant)
1799                 asset.
1800                     SellingLocationRetailers
1801                     = append(asset.
1802                         SellingLocationRetailers,
1803                         newLocation)
1804             }
1805             return "", s.SaveAsset(ctx, id,
1806                     asset)
1807         case "Consumer":
1808             if !contains(asset.Consumers,
```

```

1809             newParticipant) {
1810                 asset.Consumers = append(
1811                     asset.Consumers,
1812                     newParticipant)
1813             }
1814             return "", s.SaveAsset(ctx, id,
1815                         asset)
1816         default:
1817             return "", fmt.Errorf("invalid role
1818                         specified: %s", role)
1819         }
1820     }
1821 }
```

Listing A.5: TransferAsset Function

```

1822
1823     func (s *SmartContract) ReadAsset(ctx contractapi.
1824                                         TransactionContextInterface, id string) (*Asset,
1825                                         error) {
1826
1827         assetJSON, err := ctx.GetStub().GetState(id)
1828
1829         if err != nil {
1830
1831             return nil, fmt.Errorf("failed to
1832                 read from world state: %v", err)
1833
1834         }
1835
1836         if assetJSON == nil {
1837
1838             return nil, fmt.Errorf("the asset %s
1839                         does not exist", id)
1840
1841         }
1842
1843         var asset Asset
```

```

1836)         err = json.Unmarshal(assetJSON, &asset)
1837)         if err != nil {
1838)             return nil, err
1839)         }
1840)         if asset.Consumers == nil {
1841)             asset.Consumers = []string{}
1842)         }
1843)
1844)         return &asset, nil
1845)
1846}

```

Listing A.6: ReadAsset Function

```

1847
1848)     func (s *SmartContract) GetAllAssets(ctx contractapi.
1849)                                         .TransactionContextInterface) ([]*Asset, error) {
1850)         resultsIterator, err := ctx.GetStub().
1851)             GetStateByRange("", "")
1852)         if err != nil {
1853)             return nil, err
1854)         }
1855)         defer resultsIterator.Close()
1856)
1857)         var assets []*Asset
1858)         for resultsIterator.HasNext() {
1859)             queryResponse, err :=
1860)                 resultsIterator.Next()
1861)             if err != nil {
1862)                 return nil, err

```

```

1863:         }
1864:
1865:         var asset Asset
1866:         err = json.Unmarshal(queryResponse.
1867:                               Value, &asset)
1868:         if err != nil {
1869:             return nil, err
1870:         }
1871:         if asset.Consumers == nil {
1872:             asset.Consumers = []string{}
1873:         }
1874:         assets = append(assets, &asset)
1875:     }
1876:
1877:     return assets, nil
1878: }
1879

```

Listing A.7: GetAllAssets Function

```

1880
1881     async function main() {
1882         displayInputParameters();
1883
1884         const client = await newGrpcConnection();
1885
1886         const gateway = connect({
1887             client,
1888             identity: await newIdentity(),
1889             signer: await newSigner(),

```

```
1890)         hash: hash.sha256 ,  
1891)         evaluateOptions: () => ({ deadline:  
1892)             Date.now() + 5000 }),  
1893)         endorseOptions: () => ({ deadline:  
1894)             Date.now() + 15000 }),  
1895)         submitOptions: () => ({ deadline:  
1896)             Date.now() + 5000 }),  
1897)         commitStatusOptions: () => ({  
1898)             deadline: Date.now() + 60000 }),  
1899)     } );  
1900)  
1901)     try {  
1902)         const network = gateway.getNetwork(  
1903)             channelId);  
1904)         const contract = network.getContract  
1905)             (chaincodeName);  
1906)  
1907)         await initLedger(contract);  
1908)         await getAllAssets(contract);  
1909)         await createAsset(contract);  
1910)         await transferAssetAsync(contract);  
1911)         await readAssetByID(contract);  
1912)         await updateNonExistentAsset(  
1913)             contract);  
1914)     } finally {  
1915)         gateway.close();  
1916)         client.close();
```

```

19170 }           }

19181 }

19192 }

19203 main().catch((error) => {

19214     console.error('** FAILED to run the

19225         application:', error);

19236     process.exitCode = 1;

19247 }) ;
19258

```

Listing A.8: `main()` Function and Gateway Logic

1926 A.1 Hyperledger Fabric Deployment Instructions

1927 The following steps outline the process for deploying and testing the fish supply
 1928 chain smart contract on Hyperledger Fabric using Google Cloud Platform.

1929 A.1.1 Environment Setup

1930 1. Open GCP and access the VM instance:

- 1931 • Navigate to Console → Compute Engine → VM instances → start →
- 1932 click SSH
- 1933 • Alternatively: Virtual Machine → start → instance → SSH

1934 2. Connect to the instance:

Listing A.9: Connect to GCP VM via `gcloud`

```

1936      gcloud compute ssh instance
1937          -20250322-102900 --zone=us-central1-c
1938

```

1939 3. Navigate to the test network directory:

1940 Listing A.10: Navigate to Compose Directory

```

1941      cd ~/fabric-samples/test-network/
1942      compose
1943

```

1944 A.1.2 Network Startup and Smart Contract Deployment

1945 1. Start the Hyperledger Fabric network:

1946 Listing A.11: Start Fabric Network

```

1947      sudo docker-compose -f
1948          compose-test-net.yaml
1949      start
1950

```

1951 • Deploy the chaincode:

1952 Listing A.12: Deploy Chaincode

```

1953      cd ../
1954      ./network.sh deployCC -ccn
1955          basic -ccp ../asset-
1956          transfer-basic/chaincode-
1957          go -ccl go
1958

```

1959 • Set environment path variables:

Listing A.13: Path Environment Variables

```

1960
1961           export PATH=${PWD}/../bin:
1962                         $PATH
1963
1964           export FABRIC_CFG_PATH=${PWD}
1965                         /../config/

```

- 1966 • Configure organization environment variables:

Listing A.14: Org1 Environment Configuration

```

1967
1968     # Environment variables for
1969     Org1
1970
1971     export CORE_PEER_TLS_ENABLED
1972             =true
1973
1974     export CORE_PEER_LOCALMSPID=
1975             "Org1MSP"
1976
1977     export CORE_PEER_TLS_ROOTCERT_FILE
1978             =${PWD}/organizations/
1979             peerOrganizations/org1.
1980             example.com/peers/peer0.
1981             org1.example.com/tls/ca.
1982             crt
1983
1984     export CORE_PEER_MSPCONFIGPATH=$
1985             ${PWD}/organizations/
1986             peerOrganizations/org1.

```

```

1987           msp
1988           export CORE_PEER_ADDRESS=
1989           localhost:7051
1990

```

¹⁹⁹¹ A.1.3 Testing the Smart Contract

¹⁹⁹² 1. Initialize the ledger:

¹⁹⁹³ Listing A.15: Invoke InitLedger

```

1994           peer chaincode invoke -o localhost:7050
1995           --ordererTLSHostnameOverride orderer
1996           .example.com --tls --cafile "${PWD}/
1997           organizations/ordererOrganizations/
1998           example.com/orderers/orderer.example.
1999           com/msp/tlscacerts/tlsca.example.com-
2000           cert.pem" -C mychannel -n basic --
2001           peerAddresses localhost:7051 --
2002           tlsRootCertFiles "${PWD}/
2003           organizations/peerOrganizations/org1.
2004           example.com/peers/peer0.org1.example.
2005           com/tls/ca.crt" --peerAddresses
2006           localhost:9051 --tlsRootCertFiles "${
2007           PWD}/organizations/peerOrganizations/
2008           org2.example.com/peers/peer0.org2.
2009           example.com/tls/ca.crt" -c '{"
2010           function":"InitLedger","Args":[]}'
2011

```

2012 2. Query assets:

2013 Listing A.16: Query Fish Asset

```
2014                  # Query a specific fish asset  
2015                  peer chaincode query -C mychannel -n  
2016                  basic -c '{"Args": ["ReadAsset", "tuna1  
2017                  "]}'  
2018  
2019                  # Query all assets in the ledger  
2020                  peer chaincode query -C mychannel -n  
2021                  basic -c '{"Args": ["GetAllAssets"]}'  
2022
```

2023 3. Shut down the network:

2024 Listing A.17: Stop Fabric Network

```
2025                  sudo docker-compose -f compose-test-net  
2026                  .yaml stop  
2027
```

2028 A.1.4 Important Notes

- 2029 • Ensure proper network connectivity when working with Google Cloud Plat-
2030 form.
- 2031 • The environment variables must be set correctly for the organization context.
- 2032 • TLS certificates are required for secure communication between nodes.
- 2033 • The commands assume the blockchain network is already provisioned on
2034 GCP.

²⁰³⁵ **Appendix B**

²⁰³⁶ **Resource Persons**

²⁰³⁷ **Dr. Ricardo P. Babaran**

²⁰³⁸ Professor of Fisheries

²⁰³⁹ Institute of Marine Fisheries and Oceanology

²⁰⁴⁰ University of the Philippines Visayas

²⁰⁴¹ rpbabaran@upv.edu.ph

²⁰⁴² **Engr. Noel Lucero**

²⁰⁴³ Engineer

²⁰⁴⁴ Jagnee Fishing Corp.

²⁰⁴⁵ JagneeFishingCorp@outlook.com

²⁰⁴⁶ **Ms. Veronica Jeruta**

²⁰⁴⁷ Barangay Kagawad

²⁰⁴⁸ Sapa Barangay Hall

²⁰⁴⁹ veronicanave9@gmail.com

²⁰⁵¹ **Appendix C**

²⁰⁵² **Interview Request Letter**

²⁰⁵³ Here is the scanned copy of the letter sent to the interviewees.

Division of Physical Science and Mathematics
College of Arts and Sciences
University of the Philippines Visayas
Miag-ao, Iloilo

Dear Ma'am/Sir,

Maayong adlaw! We are a group of Computer Science students from the University of the Philippines currently working on our project entitled "SeaXChange: A Blockchain-Driven App for Tuna Supply Chain Management". As part of the development and testing phase, we would like to interview selected fishermen, suppliers, retailers and consumers here in Barangay Mat-Y to gather feedback and evaluate the usability and practicality of our application. The interview will be conducted from April 21-25, 2025.

The goal of our system is to help track tuna products from the ocean to the consumers and to improve the traceability of the tuna supply chain. We believe that the insights from the fishermen, suppliers, retailers and consumers will be extremely valuable in making it more effective and user-friendly.

We respectfully request your permission to lend us your precious time to participate in our short interviews or discussions. Rest assured that all information collected will remain confidential and will be used for academic and research purposes only.

We hope for your favorable response. Thank you for your time and support!

Sincerely,
The student researchers


Jeff Rouzel Bat-og
Student Researcher


Maxinne Gwen Cahilig
Student Researcher


Zyrex Djewel Ganit
Student Researcher

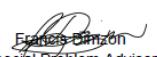

Francis Dimzon
Special Problem Adviser

Figure C.1: Scanned Interview Request Letter

2054 **Appendix D**

2055 **App Demo Documentation**

- 2056 The photographs taken show the engagement of the interviewees during the app demonstration.
- 2057



Figure D.1: Respondents (Fishermen, supplier, and retailer) from Barangay Sapa, Miagao



Figure D.2: Respondents (Retailer, and consumer) from Barangay Mat-Y, Miagao