

SEAXCHANGE: A BLOCKCHAIN DRIVEN APP FOR TUNA SUPPLY CHAIN MANAGEMENT

A Special Problem

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

Miag-ao, Iloilo

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

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

Contents

¹¹⁵	1 Introduction	¹
¹¹⁶	1.1 Overview	¹
¹¹⁷	1.2 Problem Statement	²
¹¹⁸	1.3 Research Objectives	³
¹¹⁹	1.3.1 General Objective	³
¹²⁰	1.3.2 Specific Objectives	³
¹²¹	1.4 Scope and Limitations of the Research	⁴
¹²²	1.5 Significance of the Research	⁴
¹²³		
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	23
137	2.11 Chapter Summary	25
138		
139	2.11.2 Research Gaps and Problem	25
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	114
222	D.2 Respondents (Retailer, and consumer) from Barangay Mat-Y, Miagao	115

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
435 of tuna: yellowfin *Thunnus albacares*, bigeye *Thunnus obesus*, bluefin *Thunnus*
436 *thynnus* or *Thunus orientalis*, albacore *Thunnus alalunga*, and skipjack *Katsu-*
437 *wonus pelamis* are harvested to meet the global supply chain demand which causes
438 those group of tuna fishes to be heavily exploited and threatened. The study con-

439 ducted by Paillin et al. (2022) states that there are multiple risk agents in the
440 supply chain assessment of tuna, these include the lack of standard environmental
441 management 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 iden-
447 tified areas (Macusi et al, 2023) . The implementation of closed fishing season
448 caused delay or lack of fish supply, which led to higher fish prices. The growing
449 demands and depleting population of tuna fishes coupled with the rapid increase
450 in fuel costs can have a negative impact on the future of the supply chain in tuna
451 fisheries (Mullon et al., 2017) . With factors concerning the slow decline of tuna
452 catches in the Philippines and surrounding nations, the future of the global supply
453 chain of 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
481 involved in the supply, production, distribution, and marketing of skipjack tuna
482 in Lagonoy Gulf in the Philippines. The study showcased a total of eleven inter-
483 connected 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
606 the delivery of a product or service while strongly emphasizing the customer's
607 specifications. With the increasing studies conducted and published for supply
608 chain, many companies adopted this practice for the benefit of their longevity,
609 as such the term supply chain management has come into place. The Council of
610 Supply Management Professionals or CSCMP (2024) defines supply chain man-
611 agement as the planning and management of all activities involved in sourcing
612 and procurement, conversion, and all logistics management activities; essentially,
613 supply chain management integrates supply and demand management within and
614 across the company. Supply chain management is also involved with the relation-
615 ship with collaborators and channel partners such as suppliers, intermediaries,
616 third party providers, and customers (CSCMP, 2024) .

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

2.7. OPPORTUNITIES OF BLOCKCHAIN TECHNOLOGY FOR SUPPLY CHAIN MANAGEMENT

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

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

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

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

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

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

737 **2.10 Developing a Traceability System for Tuna 738 Supply Chains**

739 The study of Kresna et al. (2017) , proposed an IT-based traceability system
740 for tuna supply chain as opposed to the traditional paper based traceability sys-

tem which has several limitations such as the potency to be manipulated, error by the human, language barrier, and physical damage. The architecture comprises several layers: infrastructure, data, application, communication, and user layers. The infrastructure layer includes computer hardware, network infrastructure, and sensing devices like CCTV, GPS, and RFID for data acquisition. The data layer serves as the system's database, featuring both a main system database and an emergency database for critical situations. The application layer consists of various modules—admin, tracing, transporter, supplier, and government—that registered actors can access through different interfaces. Finally, the user layer consists of the registered actors who utilize the system.

The journal article of Tiwari (2020) called Application of Blockchain in Agri-Food Supply Chain conducted two case studies for a blockchain driven app built for supply chain related to food, fishing, and agriculture. The first case-study is the usage and effectiveness of the Provenance system for tuna tracking certification. The objective of the Provenance system is to enhance transparency in the tuna supply chain by ensuring certification and standard compliance across all roles(e.g. supplier, retailer) in the chain. The system is built using six modular programs: registering, standards, production, manufacturing, tagging, and user-interface. The usage of blockchain in the Provenance system allows transactions to be recorded to allow shared ledger for transparency and smart contracts for secure exchanges of money or information. The usage of the Provenance system is to solve the issues encountered in the tuna fishing industry affected by various factors such as illegal, unregulated, unauthorized (IUU) fishing, fraud, and human rights abuses. The solution of the Provenance system is to allow tracking, tracing, and certification of tuna using blockchain. The Provenance system has a smart tag-

766 ging feature that allows fishermen to use SMS for digital assets on the blockchain
767 to track where the fish, in return, all supply chain stakeholders can access the
768 data that was sourced from the SMS. The second case-study is the usage of the
769 IBM Food Trust for transparency in the food supply chain. The IBM Food Trust
770 aims to solve the problems in the food supply chain, specifically in product safety.
771 Locating supply chain items in real-time using identifiers like GTIN or UPC is
772 the primary feature of the IBM Food Trust. The app also provides end-to-end
773 product provenance, real-time location and status, and facilitates rapid product
774 recalls. The IBM Food Trust also provides insights and visibility for the freshness
775 of the product to reduce losses and spoilage. Lastly, the IBM Food Trust provides
776 certifications from the information taken when handling and managing the prod-
777 ucts in the supply chain. The case studies conducted by Tiwari (2020) illustrates
778 the potential of blockchain technology in improving transparency, efficiency, and
779 ethical practices within supply chains.

780 2.11 Chapter Summary

781 2.11.2 Research Gaps and Problem

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

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

⁷⁸⁷ tion, digital literacy, and connectivity.

⁷⁸⁸ Furthermore, existing technologies have primarily focused on large-scale implementations and theoretical frameworks without adequately considering the practical implications and user experience needed for effective system integration. This study aims to address these gaps by exploring real-world challenges faced by users, especially fishermen, in adopting blockchain technology for traceability.

793 2.11.3 Summary

794 The literature reviewed highlighted the critical challenges and opportunities re-
795 garding the tuna supply chain, particularly in the areas of traceability and sustain-
796 ability. Existing supply chain technologies, especially those utilizing blockchain,
797 present solutions but also come with limitations in terms of blockchain adoption.

798 The application of blockchain technology in the tuna supply chain has shown po-
799 tential for enhancing traceability from ocean to consumer.

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

⁸⁰³ **Chapter 3**

⁸⁰⁴ **Research Methodology**

⁸⁰⁵ This chapter outlines a clear and detailed description of the research methods and
⁸⁰⁶ processes used in the development and evaluation of SeaXChange: A Blockchain
⁸⁰⁷ Driven App for Tuna Supply Chain Management. The algorithms, systems, theo-
⁸⁰⁸ ries, framework and models are described in detail in which this chapter establishes
⁸⁰⁹ the foundation of this study .This chapter also explains the data collection method
⁸¹⁰ used ensuring the validity and reliability of the results.In addition, the chapter
⁸¹¹ discusses the considerations and potential limitations of this study. Overall, this
⁸¹² will serve as a guide for the readers in understanding the structured process of
⁸¹³ developing the SeaXChange.

⁸¹⁴ **3.1 Software Development**

⁸¹⁵ Scrum is a framework within the Agile development that prioritizes flexibility.
⁸¹⁶ It is an iterative software development approach that lets a project be broken

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

822 **3.2 Research Activities**

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

830 **3.2.1 Feedback Collection Method**

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

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

845 3.2.2 Data Gathering

846 • Primary Data:

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

852 • Secondary Data:

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

857 3.2.3 Designing and Developing the System

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

861 2. **Technology Stack:**

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

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

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

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

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

878 3. **Blockchain Development Platform:**

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

884 **3.2.4 Implementing Algorithms and Services**

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

889 **1. Consensus Algorithm**

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

897 **2. Cryptographic Algorithm**

898 The project employed several cryptographic algorithms to ensure security
899 and privacy. These cryptographic data served as digital signatures and iden-
900 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

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

929 **5. Endorsement Policy**

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

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

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

944 **3.2.5 Modeling the System Architecture**

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

949 • **Blockchain Architecture**

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

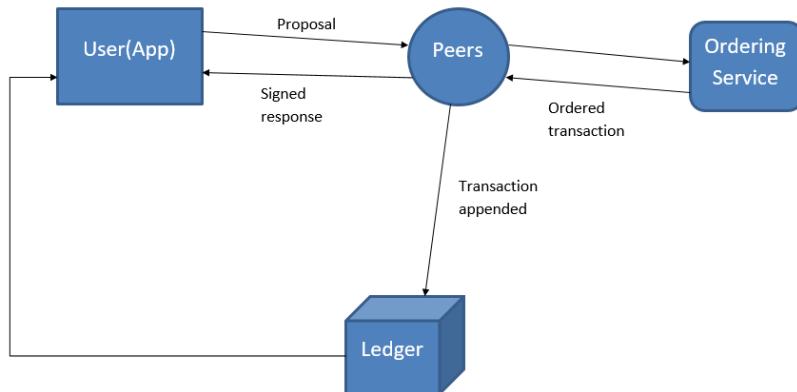


Figure 3.1: Blockchain Architecture of SeaXChange

960 • **Overall System Architecture**

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

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

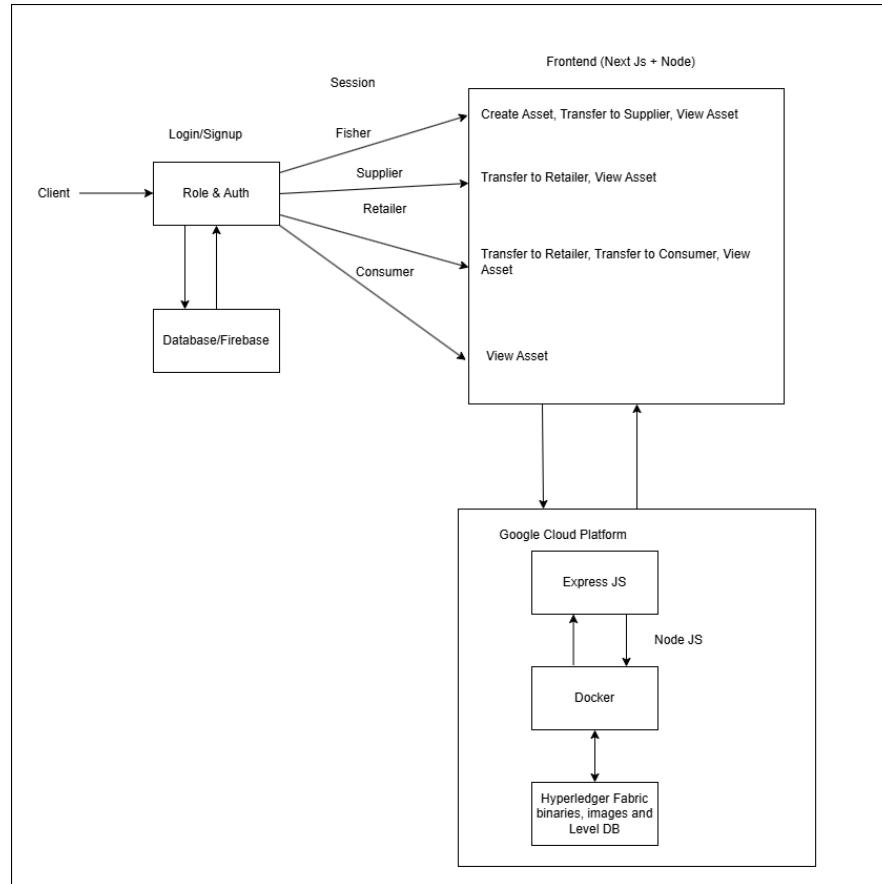


Figure 3.2: Overall System Architecture of SeaXChange

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

977 **1. Fisher**

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

983 **2. Supplier**

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

987 **3. Retailer**

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

992 **4. Consumer**

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

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

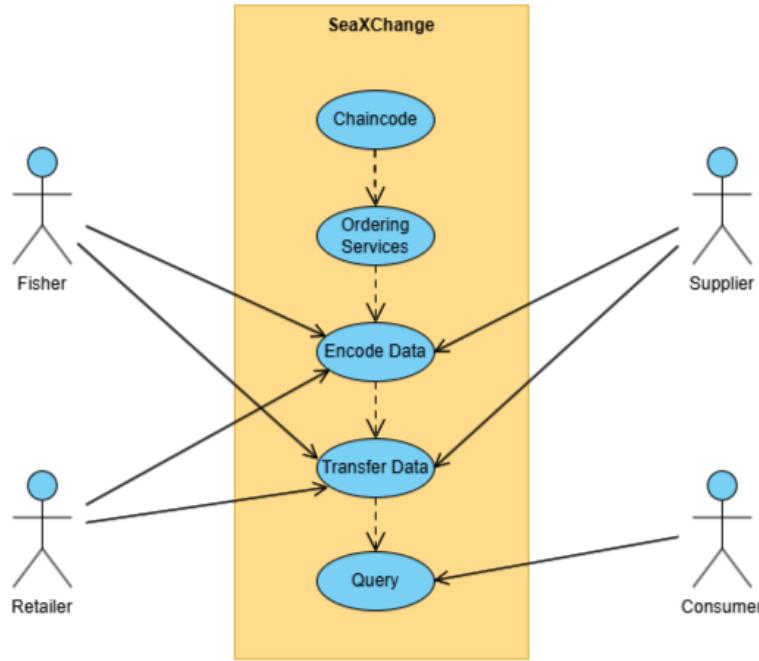


Figure 3.3: Use case diagram for SeaXChange.

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

¹⁰⁰⁷ **Chapter 4**

¹⁰⁰⁸ **Results and Discussions**

¹⁰⁰⁹ **4.1 Overview**

¹⁰¹⁰ This chapter presents the results of the system. Included in this chapter are
¹⁰¹¹ screenshots and the discussion of results. The tuna supply chain management
¹⁰¹² smart contract on Hyperledger Fabric has been initiated and tested within a con-
¹⁰¹³ trolled blockchain environment. Results indicated that the system was function-
¹⁰¹⁴ ally robust and reliable, having managed assets, transaction integrity, and the
¹⁰¹⁵ ability to query and update the ledger in the blockchain. This chapter presents
¹⁰¹⁶ the details of the major steps executed during the process, results for those steps,
¹⁰¹⁷ and the current status of the system's operations.

1018 **4.2 Smart Contract Deployment and Installa-**

1019 **tion**

1020 **4.2.1 Hyperledger Fabric Prerequisites**

1021 Before executing a smart contract framework and blockchain system, it is cru-
1022 cial to first install and set up the necessary tools and technologies. This includes
1023 setting up Hyperledger Fabric, which involves installing the Fabric binaries, con-
1024 figuring the network, and ensuring all necessary dependencies like Docker, Docker
1025 Compose, and Node.js are installed and properly configured. Additionally, setting
1026 up the required certificates, defining the channel configurations, and ensuring that
1027 peer nodes and orderers are correctly connected and synchronized are all essential
1028 steps in preparing the environment for blockchain and smart contract operations.

1029 • **Software Requirements:**

- 1030 – **Docker and Docker Compose:** Hyperledger Fabric needs to have
1031 Docker installed and running on the system. Docker is required to run
1032 the peer and ordering services of the blockchain system.
- 1033 – **Node.js:** Required for the Fabric SDK for client application integra-
1034 tion with JavaScript libraries such as react.
- 1035 – **Go:** Ensure Go is installed, and the GOPATH environment variable
1036 is set up. This is essential for building and running chaincode(smart
1037 contract) written in Go.
- 1038 – **Fabric Samples:** Clone the official Hyperledger Fabric's fabric-samples
1039 repository from GitHub:

```
1040      git clone -b release-2.4 --single-branch  
1041          https://github.com/hyperledger/fabric-samples  
1042      cd fabric-samples/test-network  
1043
```

1044 – **Binaries and Docker Images:**

```
1045 curl -sSL https://bit.ly/2ysbOFE | bash -s
```

```
1046
```

1047 • **Network Setup:**

- ```
1048 – Run the test-network script to start the Hyperledger Fabric test net-
1049 work:
```

```
1050 ./network.sh up
```

```
1051
```

1052 This script starts a peer, an ordering service, and a CA (Certificate  
1053 Authority) on the local machine.

- ```
1054 – After starting the network to docker in the same directory (test-network),  
1055     a channel must be created:
```

```
1056     ./network.sh createChannel
```

```
1057
```

1058 • **Deploying Chaincode (Smart Contract):**

- ```
1059 – Step 1:
```

```
1060 export PATH=${PWD}/../bin:$PATH
```

```
1061
```

1062 – Step 2:

```
1063 export FABRIC_CFG_PATH=$PWD/./config/
```

```
1064
```

1065 – Step 3:

```
1066 export CORE_PEER_TLS_ENABLED=true
1067 export CORE_PEER_LOCALMSPID="Org1MSP"
1068 export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations
1069 /peerOrganizations/org1.example.com/peers/peer0.org1.example.com
1070 /tls/ca.crt
1071 export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations
1072 /peerOrganizations/org1.example.com/users
1073 /Admin@org1.example.com/msp
1074 export CORE_PEER_ADDRESS=localhost:7051
1075
1076
```

### 1077 4.2.2 Invoking the Blockchain System

1078 After setting up the prerequisites, including Docker containers, the test network,  
 1079 and chaincode, the tuna supply chain system can now be invoked for creating,  
 1080 transferring, and querying tuna assets. The figures provided below demonstrate  
 1081 the processes involved in invoking the blockchain system.

## 4.2. SMART CONTRACT DEPLOYMENT AND INSTALLATION

45

```
[root@ryrex2082 LAPTOP-Q93UQWB:~/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": "Reyes"}, {"ID": "tuna3", "Species": "Bluefin", "Weight": 5.5, "CatchLocation": "Antique", "CatchDate": "2024-12-03", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierZ", "Retailer": "Cruz"}, {"ID": "tuna4", "Species": "Albacore", "Weight": 5.5, "CatchLocation": "Antique", "CatchDate": "2024-12-04", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierA", "Retailer": "Cruz"}, {"ID": "tuna5", "Species": "Albacore", "Weight": 6, "CatchLocation": "Davao", "CatchDate": "2024-12-05", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "NA", "Retailer": "MA"}, {"ID": "tuna6", "Species": "Albacore", "Weight": 6, "CatchLocation": "Davao", "CatchDate": "2024-12-05", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierX", "Retailer": "Retailery"}]
ryrex2082@LAPTOP-Q93UQWB:~/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-network$
```

Figure 4.1: Query Smart Contract: Check assets

### 1082 • Adding new tuna assets:

1083 Here, a new tuna asset is created and registered on the blockchain. This in-  
 1084 volves invoking the smart contract to add a new entry, which includes details  
 1085 such as the type of tuna, quantity, and any other relevant information. This  
 1086 step ensures that newly caught or acquired tuna can be tracked throughout  
 1087 the supply chain.

```
[root@ryrex2082 LAPTOP-Q93UQWB:~/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 -peerAddresses localhost:7051 --tlsRootCertFiles $PWD/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddressTLSHeader "SkipProof" -f "2024-12-06T10:00:00Z" -c '{"function": "CreateAsset", "Args": [{"ID": "tuna7", "Species": "Skipjack", "Weight": 6.5, "CatchLocation": "Antique", "CatchDate": "2024-12-06", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierX", "Retailer": "Retailery"}]} >>> Chaincode invoke successful, result: status:200
ryrex2082@LAPTOP-Q93UQWB:~/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-network$
```

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

### 1088 • Check all assets after adding a new tuna asset:

1089 After adding a new tuna asset, the smart contract is queried again to verify  
 1090 that the asset has been successfully added. This step confirms that the new  
 1091 asset is part of the current inventory and that no discrepancies exist in the  
 1092 recorded data.

```
[root@ryrex2082 LAPTOP-Q93UQWB:~/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": "Reyes"}, {"ID": "tuna3", "Species": "Bluefin", "Weight": 5.5, "CatchLocation": "Antique", "CatchDate": "2024-12-03", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierZ", "Retailer": "Cruz"}, {"ID": "tuna4", "Species": "Albacore", "Weight": 5.5, "CatchLocation": "Antique", "CatchDate": "2024-12-04", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierA", "Retailer": "Cruz"}, {"ID": "tuna5", "Species": "Albacore", "Weight": 6, "CatchLocation": "Davao", "CatchDate": "2024-12-05", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "NA", "Retailer": "MA"}, {"ID": "tuna6", "Species": "Albacore", "Weight": 6, "CatchLocation": "Davao", "CatchDate": "2024-12-05", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "SupplierX", "Retailer": "Retailery"}, {"ID": "tuna7", "Species": "Skipjack", "Weight": 6.5, "CatchLocation": "Antique", "CatchDate": "2024-12-06", "FishingMethod": "Longline", "Vessel": "Jagnew", "Supplier": "NA", "Retailer": "MA"}]
```

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

### 1093 • Transfer tuna asset to Supplier:

1094 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.

```
1095 Error: error getting broadcast client: orderer client failed to connect to orderer.example.com:7050: failed to create new connection: context deadline exceeded
1096 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"
1097 args":["tuna6","Supplier","SupplierA"]
1098
```

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

#### 1099 • Check the updated tuna asset:

1100 After the transfer, the smart contract is queried once more to check if the  
 1101 asset details have been updated correctly. This step verifies that the asset's  
 1102 new owner is now the supplier and that all relevant information is correctly  
 1103 updated on the blockchain.

```
1104 rsys2002@LAPTOP-Q93QUN8:/mnt/c/SpecialProblem/hyperledger-fabric3/fabric-samples/test-network$ peer chaincode query -C "basic-peer" -n basic -c "[{"Args": ["TransferAsset", "tuna6", "Supplier", "SupplierA"]}]"
1105 Error: error getting broadcast client: orderer client failed to connect to orderer.example.com:7050: failed to create new connection: context deadline exceeded
1106 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"
1107 args:["tuna6","Supplier","SupplierA"]
1108
```

Figure 4.5: Query Smart Contract: Check asset after transfer

#### 1104 • Transfer tuna asset to Retailer:

1105 Similar to the supplier transfer, this step involves transferring the tuna asset  
 1106 from the supplier to a retailer. The smart contract facilitates this transfer,  
 1107 ensuring that ownership is correctly updated and that the retailer has control  
 1108 over the tuna asset. This step is crucial for the supply chain as it moves the  
 1109 tuna from bulk supply to retail.

```
1110 [{"ID": "tuna6", "Species": "Skipjack", "Weight": 6, "CatchLocation": "2024-12-05", "CatchDate": "Antique", "FishingMethod": "Longline", "Vessel": "Jagney", "Supplier": "SupplierA", "Retailer": "RetailerA"}, {"ID": "tuna6", "Species": "Skipjack", "Weight": 6, "CatchLocation": "2024-12-05", "CatchDate": "Antique", "FishingMethod": "Longline", "Vessel": "Jagney", "Supplier": "SupplierA", "Retailer": "RetailerB"}]
1111 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"
1112 args:["tuna6","Supplier","SupplierA"]
1113
```

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

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

1110     • **Check the updated tuna asset:**

1111     After the transfer to the retailer, another query is made to verify the updated  
1112     asset details. This step ensures that the transaction was successful and that  
1113     the retailer now has ownership of the tuna asset. It confirms that the asset  
1114     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

1115     • **Query Smart Contract and check updated assets:**

1116     The final step involves querying the smart contract to get a complete overview  
1117     of all the assets in the supply chain. This includes all tuna assets from fishing  
1118     to retail, allowing stakeholders to monitor and manage inventory effectively.  
1119     It provides traceability in the supply chain, helping to maintain freshness  
1120     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": "Yellowtail", "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

<sub>1121</sub> **4.3 Backend Security Analysis (Hyperledger Fab-**

<sub>1122</sub> **ric on GCP)**

<sub>1123</sub> **4.3.1 System Architecture and Deployment Overview**

<sub>1124</sub> The backend of the system's tuna assets was developed using a containerized  
<sub>1125</sub> Hyperledger Fabric deployed on Google Cloud Platform (GCP). The network of  
<sub>1126</sub> Hyperledger Fabric consists of a peer node, an ordering node, and Certificate  
<sub>1127</sub> Authorities (CAs).

<sub>1128</sub> **4.3.2 Blockchain Network Security**

<sub>1129</sub> The blockchain network leverages Hyperledger Fabric's security model to ensure  
<sub>1130</sub> authenticated transactions and controlled access. A Membership Service Provider  
<sub>1131</sub> (MSP) manages identities and issues certificates based on a Public Key Infrastruc-  
<sub>1132</sub> ture (PKI) model, ensuring that only verified participants can interact with the  
<sub>1133</sub> network.

<sub>1134</sub> Key security features include:

<sub>1135</sub> **Channel Privacy**

<sub>1136</sub> Channels act as private communication subnets, isolating transaction data so that  
<sub>1137</sub> only authorized organizations can access and process specific transactions.

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

#### **Policies and Access Control**

1138 Policies, including endorsement policies and access control lists (ACLs), govern  
1139 how transactions are validated, how channel resources are accessed, and how  
1140 changes to the network are approved. Endorsement policies specifically define  
1141 which peer nodes must approve a transaction before it is committed to the ledger.  
1142

#### **Secure Communication**

1143 Transport Layer Security (TLS) is enforced across node communications to protect  
1144 data in transit. Mutual TLS is used for operational endpoints like monitoring  
1145 services.  
1146

#### **Identity and Role Management**

1147 Every network participant—peer nodes, orderer nodes, client applications (SeaX-  
1148 Change Web Application)—has a cryptographically verifiable identity, with roles  
1149 defined within the framework to control access and permissions within channels.  
1150

#### **Hardware Security Modules (HSMs)**

1151 Critical cryptographic operations, such as signing transactions under the blockchain  
1152 assets invocation, can optionally be handled by HSMs to secure private keys out-  
1153 side of the software environment.  
1154

1155 These layered mechanisms collectively ensure the confidentiality, integrity, and

1156 authenticity of transactions in the blockchain network.

### 1157 4.3.3 Smart Contract Automated Test Result

1158 To validate the security and functionality of the deployed smart contracts on the  
1159 Hyperledger Fabric network, an automated testing script (app.js) under asset-  
1160 transfer-basic directory was executed. The script interacted with the blockchain  
1161 network through the gateway application, utilizing the defined channel (mychan-  
1162 nel) and chaincode (basic). The automated tests performed the following opera-  
1163 tions:

#### 1164 InitLedger Transaction

1165 The ledger was initialized by creating a predefined set of tuna asset entries. The  
1166 transaction was successfully committed, confirming the proper initialization of  
1167 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
```

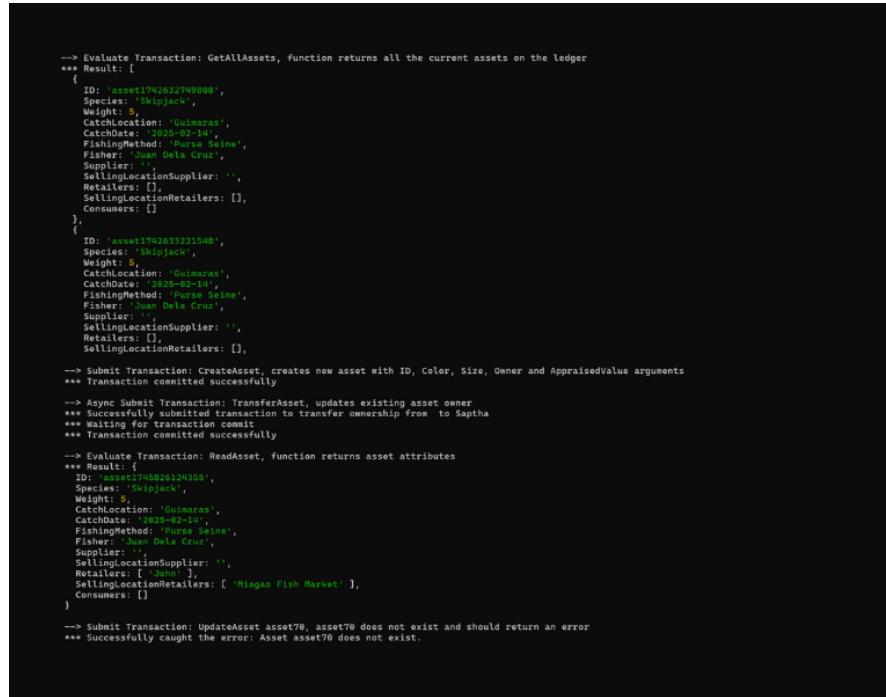
Figure 4.9: Initialization Confirmation of the Ledger

#### 1168 GetAllAssets Query

1169 A query operation retrieved all existing assets recorded on the ledger. The results  
1170 displayed multiple tuna asset entries with details such as species, weight, catch

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

1171 location, catch date, fishing method, fisher, supplier, retailers, selling locations,  
1172 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 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: 'asset174636124355',
 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

#### 1173 CreateAsset Transaction

1174 A new asset was successfully created and appended to the blockchain. The trans-  
1175 action was committed without errors, validating the chaincode's ability to handle  
1176 new data insertion.

#### 1177 TransferAsset Transaction

1178 Ownership transfer of an existing asset was simulated. The transaction was suc-  
1179 cessfully submitted and committed, demonstrating the correct application of asset

1180 updates in the blockchain ledger.

### 1181 **ReadAsset Query**

1182 The updated asset was retrieved and verified to ensure the correctness of the  
1183 transfer. The retrieved asset data reflected the changes made through the previous  
1184 transaction, confirming data consistency.

### 1185 **UpdateAsset Error Handling**

1186 An attempt to update a non-existent asset (asset70) was performed to test the  
1187 smart contract's error-handling mechanism. The application correctly caught and  
1188 reported the error, verifying that improper transactions are adequately handled  
1189 and rejected.

### 1190 **Summary of Results**

1191 All positive transactions (initialization, creation, transfer, and reading) were suc-  
1192 cessfully executed and committed. The smart contract exhibited robust error  
1193 handling during invalid operations. Endorsement policies and Membership Ser-  
1194 vice Provider (MSP) enforcement ensured transaction authenticity and integrity  
1195 during the process. These tests confirm the functional reliability and transac-  
1196 tion security of the smart contracts used within the tuna supply chain blockchain  
1197 network.

<sub>1198</sub> **4.3.4 GCP Infrastructure Security**

<sub>1199</sub> The Hyperledger Fabric network deployment on Google Cloud Platform (GCP)  
<sub>1200</sub> was secured by leveraging multiple layers of Google's infrastructure security model,  
<sub>1201</sub> following best practices in network, identity, and data protection.

<sub>1202</sub> **Firewall Rules and Network Control**

<sub>1203</sub> Only essential ports (e.g., 7051 for peer communication and 7050 for the ordering  
<sub>1204</sub> service) were opened to minimize network exposure. GCP's VPC firewall rules,  
<sub>1205</sub> ingress and egress controls, and options like VPC Service Controls help further  
<sub>1206</sub> isolate services and prevent unauthorized access. Traffic between virtual machines  
<sub>1207</sub> and Google APIs is securely routed through Google's internal network infrastruc-  
<sub>1208</sub> ture using Private Google Access when available.

<sub>1209</sub> **IAM Roles and Access Management**

<sub>1210</sub> The principle of least privilege was enforced by assigning minimal permissions to  
<sub>1211</sub> users and services through GCP Identity and Access Management (IAM). Access  
<sub>1212</sub> decisions involve authentication, authorization checks, and enforcement of policies  
<sub>1213</sub> through centralized services, helping reduce the risk of unauthorized actions or  
<sub>1214</sub> privilege escalation.

**1215    Encryption**

1216    GCP ensures that data is encrypted both at rest and in transit by default. Storage  
1217    systems use multiple layers of encryption, with cryptographic keys managed by  
1218    Google. For additional control, Cloud Key Management Service (KMS) enables  
1219    customers to manage their own encryption keys. Data in transit between services  
1220    is protected using Application Layer Transport Security (ALTS), and all external  
1221    communication with Google services is encrypted via TLS.

**1222    Access Control**

1223    Access to virtual machines and services was restricted using secure access methods.  
1224    Identity-Aware Proxy (IAP) or VPN connections were employed to safeguard SSH  
1225    and administrative access. GCP's zero-trust model emphasizes verifying identity  
1226    and device security rather than relying solely on network location, aligning with  
1227    best practices for modern infrastructure protection.

**1228    Infrastructure and Operational Security**

1229    GCP's underlying infrastructure benefits from Google's proprietary hardware de-  
1230    signs, including the Titan security chip, secure boot mechanisms, and service iden-  
1231    tity enforcement. Google's physical data centers use multi-layered defenses such  
1232    as biometrics and intrusion detection systems. Operational security practices, in-  
1233    cluding binary authorization and extensive monitoring, reduce insider risks and  
1234    enforce software integrity throughout the lifecycle.

<sub>1235</sub> By deploying the blockchain network on GCP, the project leveraged a highly se-  
<sub>1236</sub> cure environment, benefiting from Google's layered security architecture across  
<sub>1237</sub> networking, identity management, encryption, access control, and operational  
<sub>1238</sub> practices.

#### <sub>1239</sub> 4.3.5 Threat Model and Mitigations

<sub>1240</sub> 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

<sub>1241</sub>

## <sub>1242</sub> 4.4 Mockups

<sub>1243</sub> The mockups represent the preliminary design for the SeaXChange web applica-  
<sub>1244</sub> tion, created using Figma to facilitate collaboration and incorporate feedback  
<sub>1245</sub> efficiently. The visual design features a teal-based color scheme to evoke an oceanic  
<sub>1246</sub> theme, aligning with the app's focus on tuna products supply chain. This aesthetic  
<sub>1247</sub> choice reinforces the app's identity and enhances user engagement.

<sub>1248</sub> Upon launching the app, users are first directed to the Login or Sign-Up page,  
<sub>1249</sub> where authentication is required to access any data. This ensures security and

1250 role-specific access within the blockchain system.

1251 Once authenticated, users are redirected to their personalized homepages, which  
1252 include their profile information and a dashboard showing relevant tuna product  
1253 assets. The user experience is role-dependent:

1254 • All users can view available assets on the blockchain.

1255 • Fishermen are the only users who can create new assets, representing newly  
1256 caught tuna.

1257 • Suppliers and Retailers have the ability to pass on assets down the supply  
1258 chain, updating the product's status, location, or freshness.

1259 This role-based structure ensures traceability, accountability, and clarity across  
1260 the supply chain, while maintaining a clean and intuitive interface tailored to  
1261 each user type.

#### 4.4. MOCKUPS

57

**Login Page**

SeaXChange

Email:

Password:

**Log In**

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

**Fisher Homepage**

SeaXChange

Logout Tuna 32

+ ADD CATCH

|                                   |                                   |                                   |
|-----------------------------------|-----------------------------------|-----------------------------------|
| Picture                           | Picture                           | Picture                           |
| TUNA1<br>Dec 1, 2024<br>Available | TUNA1<br>Dec 1, 2024<br>Available | TUNA1<br>Dec 1, 2024<br>Available |
| Picture                           | Picture                           | Picture                           |
| TUNA1<br>Dec 1, 2024<br>Available | TUNA1<br>Dec 1, 2024<br>Available | TUNA1<br>Dec 1, 2024<br>Available |

**Fisher Add Catch Page**

SeaXChange

Logout Tuna 32

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** **MARK AS SOLD**

**Fisher Add Catch Page 2**

SeaXChange

Logout Tuna 32

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** **MARK AS SOLD**

You won't be able to edit the tuna details once it's sent.

**Supplier Homepage**

SeaXChange

Logout Tuna 32

|                                   |                                   |                                   |
|-----------------------------------|-----------------------------------|-----------------------------------|
| Picture                           | Picture                           | Picture                           |
| TUNA1<br>Dec 1, 2024<br>Available | TUNA1<br>Dec 1, 2024<br>Available | TUNA1<br>Dec 1, 2024<br>Available |
| Picture                           | Picture                           | Picture                           |
| TUNA1<br>Dec 1, 2024<br>Available | TUNA1<br>Dec 1, 2024<br>Available | TUNA1<br>Dec 1, 2024<br>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

Logout Tuna 32

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** **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

## <sup>1262</sup> 4.5 Operational Flow of the Web Application

<sup>1263</sup> This section will discuss the flow in using the SeaXChange Web Application. It  
<sup>1264</sup> will show the respective interface for every page and how the users can interact  
<sup>1265</sup> with it.

### <sup>1266</sup> 4.5.1 Landing Page

<sup>1267</sup> Users are be greeted with the landing page, where it shows a ocean visuals and  
<sup>1268</sup> a tagline “Discover the Journey your tuna made from the ocean to your dinner  
<sup>1269</sup> plate”. Users are given the option to Login, where they are redirected to the login  
<sup>1270</sup> page or Get Started, where they are redirected to the sign up page.



Figure 4.12: Landing Page

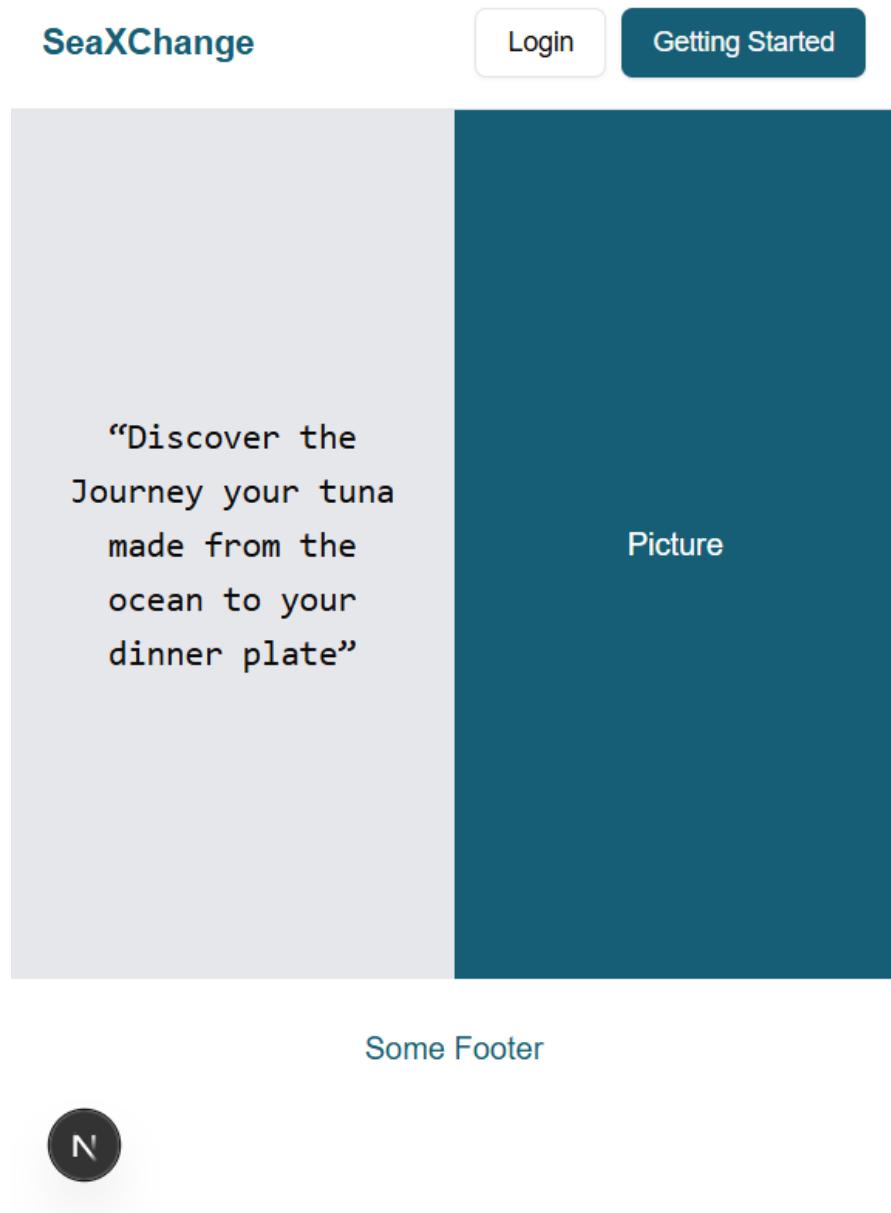


Figure 4.13: Mobile View: Landing Page

1271 **4.5.2 Sign Up Page**

1272 First time users will be required to create an account. They are to provide their  
1273 name, email and password. For their user type, there are four roles to choose  
1274 from: Fisher, Supplier, Retailer and Consumer.

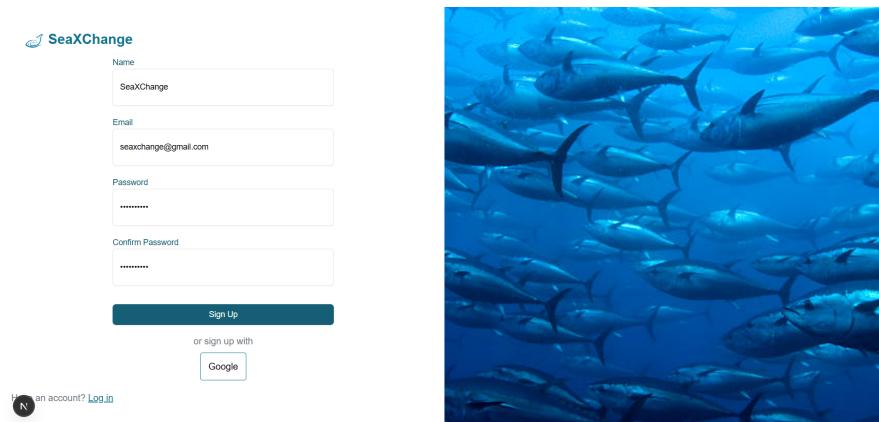


Figure 4.14: SignUp Page

1275 **4.5.3 Login Page**

1276 Returning user are required to login with their email and password.

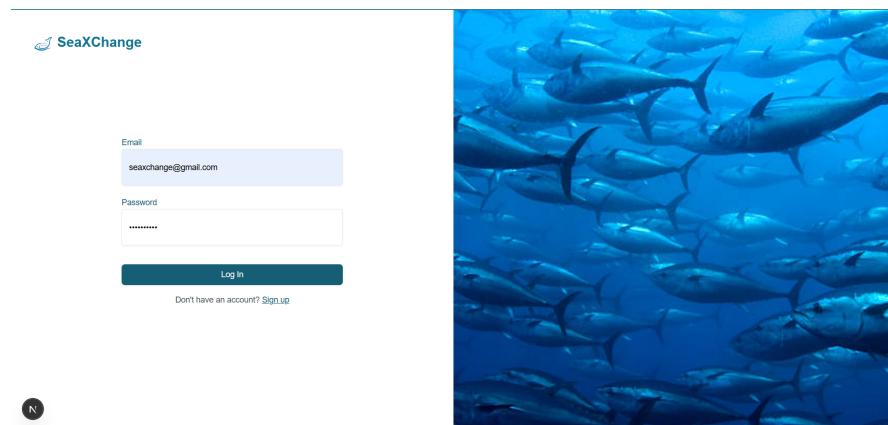


Figure 4.15: LogIn Page

<sup>1277</sup> **4.5.4 Homepage**

<sup>1278</sup> Each user type has their own respective homepages and features.

- <sup>1279</sup> • **Fisher** Fishers can add a fish catch using the "Add catch" button, where
- <sup>1280</sup> they are to input the species of the fish, weight in kg, catch location, catch
- <sup>1281</sup> date, fishing method used and vessel name. The remaining text fields like
- <sup>1282</sup> the Supplier name, Retailer name and Consumer name are left null and
- <sup>1283</sup> cannot be edited because they will be filled out by the other users receiving
- <sup>1284</sup> the tuna asset. Users can send a tuna asset to the Supplier. Users can also
- <sup>1285</sup> browse existing tuna assets.

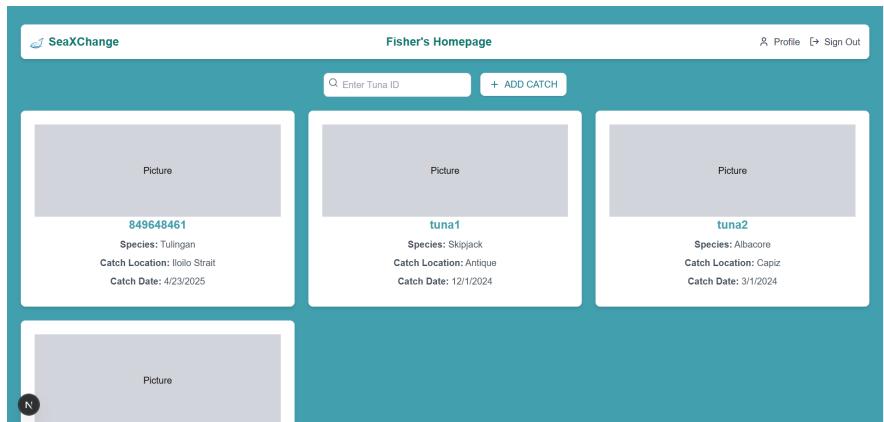


Figure 4.16: Fisher Homepage

- <sup>1286</sup> • **Supplier** Suppliers can browse existing tuna assets. Upon clicking a tuna
- <sup>1287</sup> asset, the user can only edit the Supplier text field. They can send the tuna
- <sup>1288</sup> asset to the Retailer.

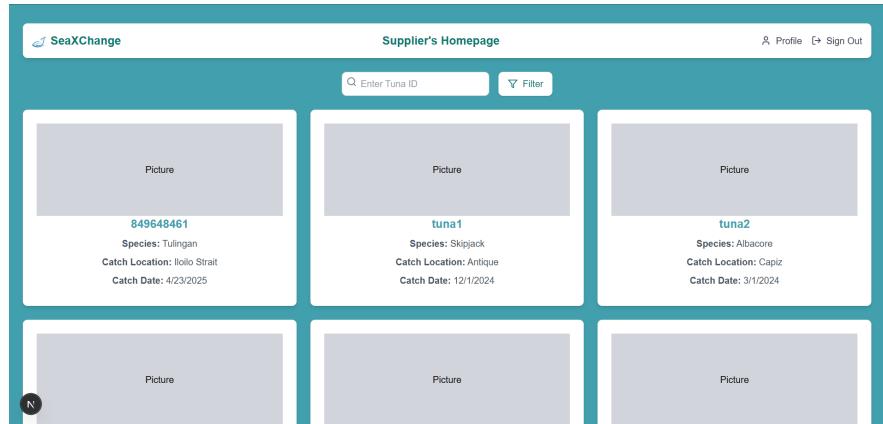


Figure 4.17: Supplier Homepage

- 1289 • **Retailer** Retailers can browse existing tuna assets and can send it to the  
1290 Consumer.

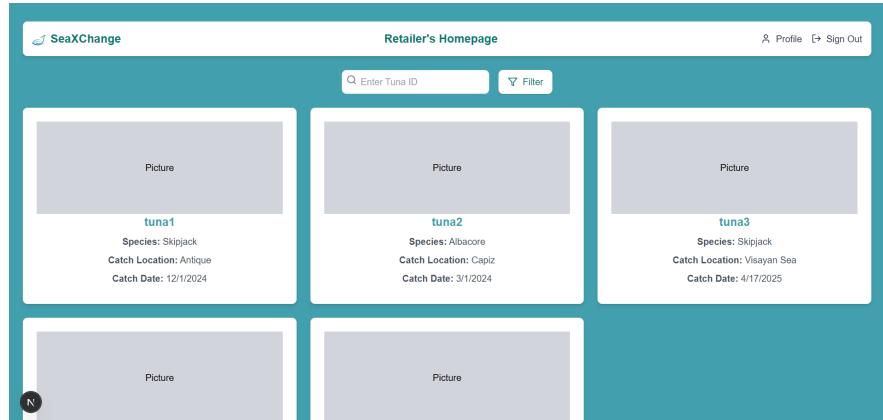


Figure 4.18: Retailer Homepage

- 1291 • **Consumer** Consumers can only view the tuna asset and cannot edit any-  
1292 thing else

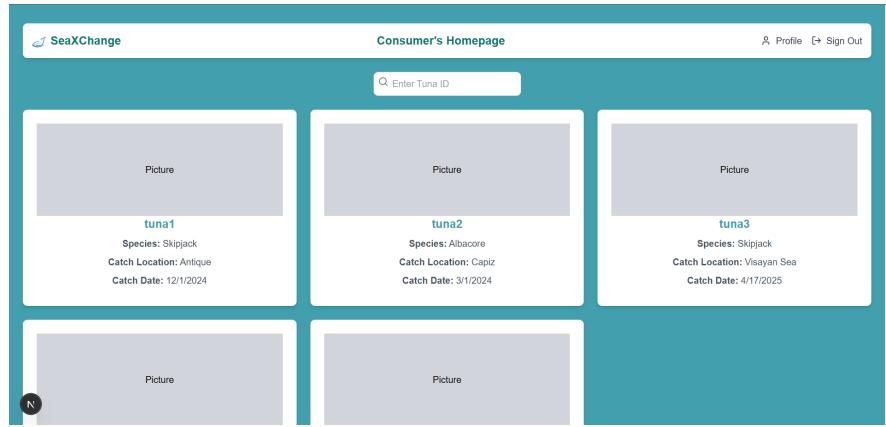


Figure 4.19: Consumer Homepage

### <sup>1293</sup> 4.5.5 Profile

- <sup>1294</sup> The user's profile information is shown on the homepage through a pop-up. It  
<sup>1295</sup> shows the user's name and role.

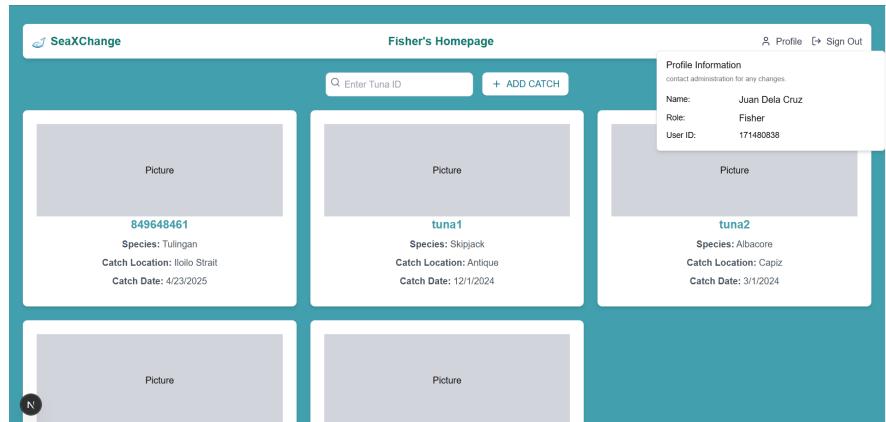


Figure 4.20: View Profile

### <sup>1296</sup> 4.5.6 Logout

- <sup>1297</sup> Users can logout of their accounts and is redirected to the Signup Page.

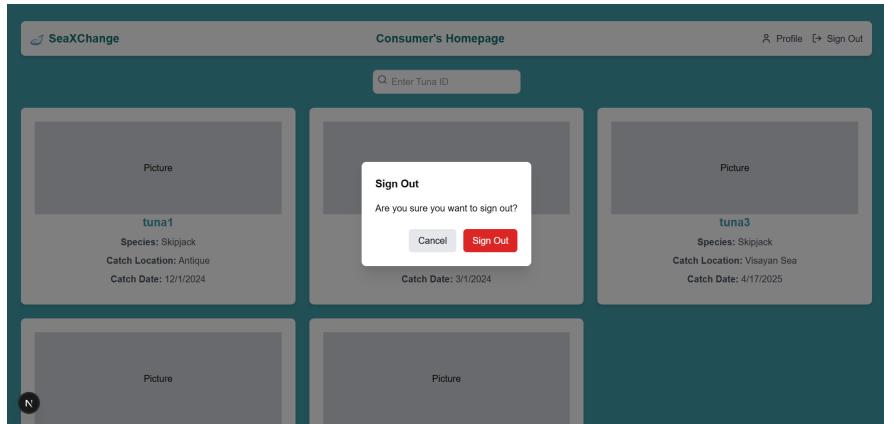


Figure 4.21: Log Out

## 1298 4.6 System Discussion

1299 After modifying the Hyperledger Fabric smart contract to assess necessary pro-  
1300 cesses involved in the tuna supply chain, the blockchain is ready to be invoked  
1301 wherein the smart contract can be activated. To start, a new tuna asset is added  
1302 and registered to the blockchain. Each tuna asset has its attributes or details.  
1303 Before proceeding to the transfer of tuna asset, the smart contract is queried to  
1304 verify if the creation of the asset is successful and if it is part of the current in-  
1305 ventory. After that, the tuna asset can be transferred from fisher to supplier and  
1306 the asset's owner is updated. The smart contract is queried again to verify if the  
1307 asset details have been updated successfully. With the same process, the tuna as-  
1308 set is transferred from supplier to retailer using the smart contract and the owner  
1309 is updated again. To ensure that the asset details are successfully updated, the  
1310 smart contract is queried again. The final step is to query the smart contract to  
1311 show the overview of all the assets in the supply chain. With this, it can be seen  
1312 all the tuna assets from fishing to retail. Overall, the steps and process provides

<sup>1313</sup> transparency and traceability in the tuna supply chain.

## <sup>1314</sup> 4.7 User Demonstration and Feedback Results

### <sup>1315</sup> 4.7.1 Demo Setup and Scenario

<sup>1316</sup> During the demonstration of the system, the participants had a brief introduction  
<sup>1317</sup> of the key functionalities of the SeaXChange app. They were shown how to  
<sup>1318</sup> create an account, input and send tuna assets from one stakeholder to another.

<sup>1319</sup> Participants were also shown how real-time updates were reflected on the app.

<sup>1320</sup> Finally, they were introduced on how to view transaction histories and traceability  
<sup>1321</sup> information on each tuna asset. Throughout the demonstration, participants  
<sup>1322</sup> were encouraged to ask questions and provide feedback on the usability and func-  
<sup>1323</sup> tionality of the system. After the demonstration, they were given feedback forms  
<sup>1324</sup> in order to assess the SeaXChange app.

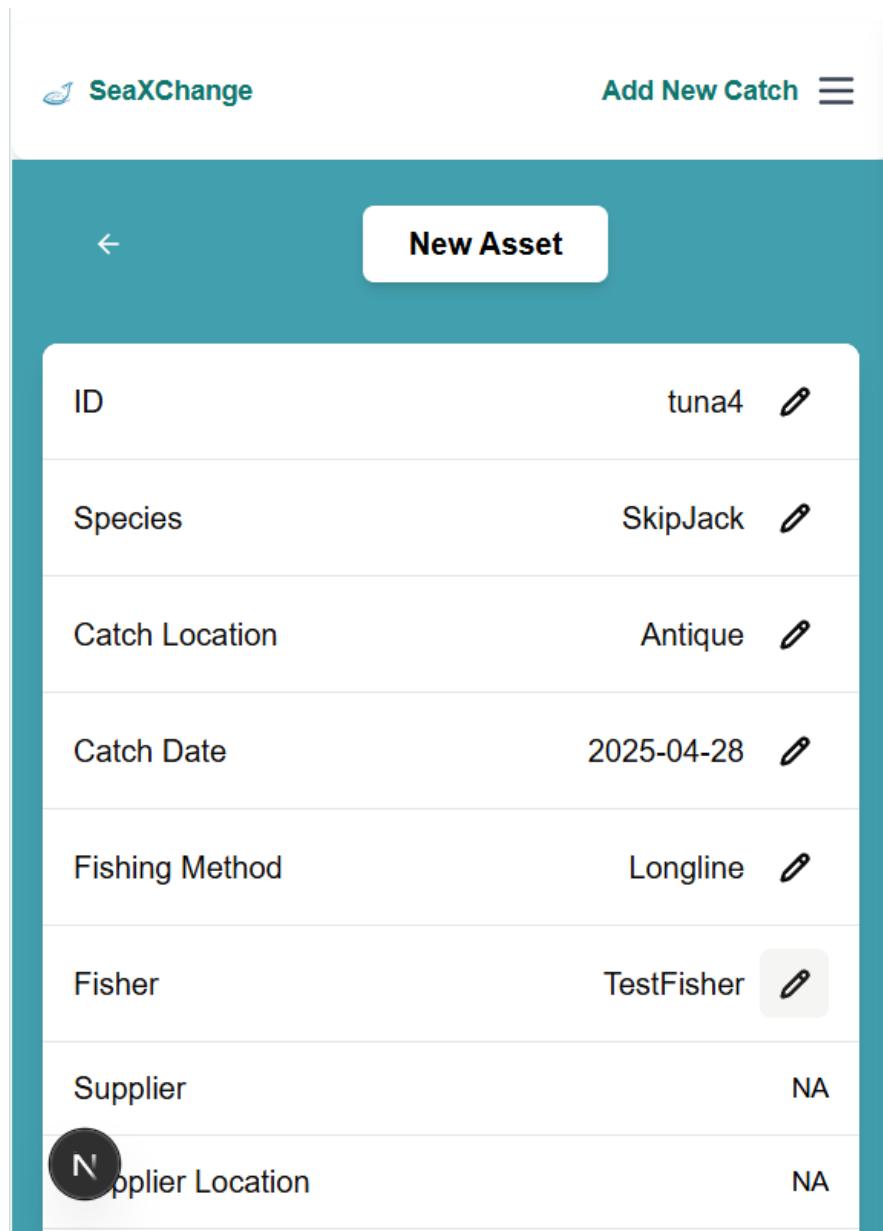


Figure 4.22: Add Catch (Asset)

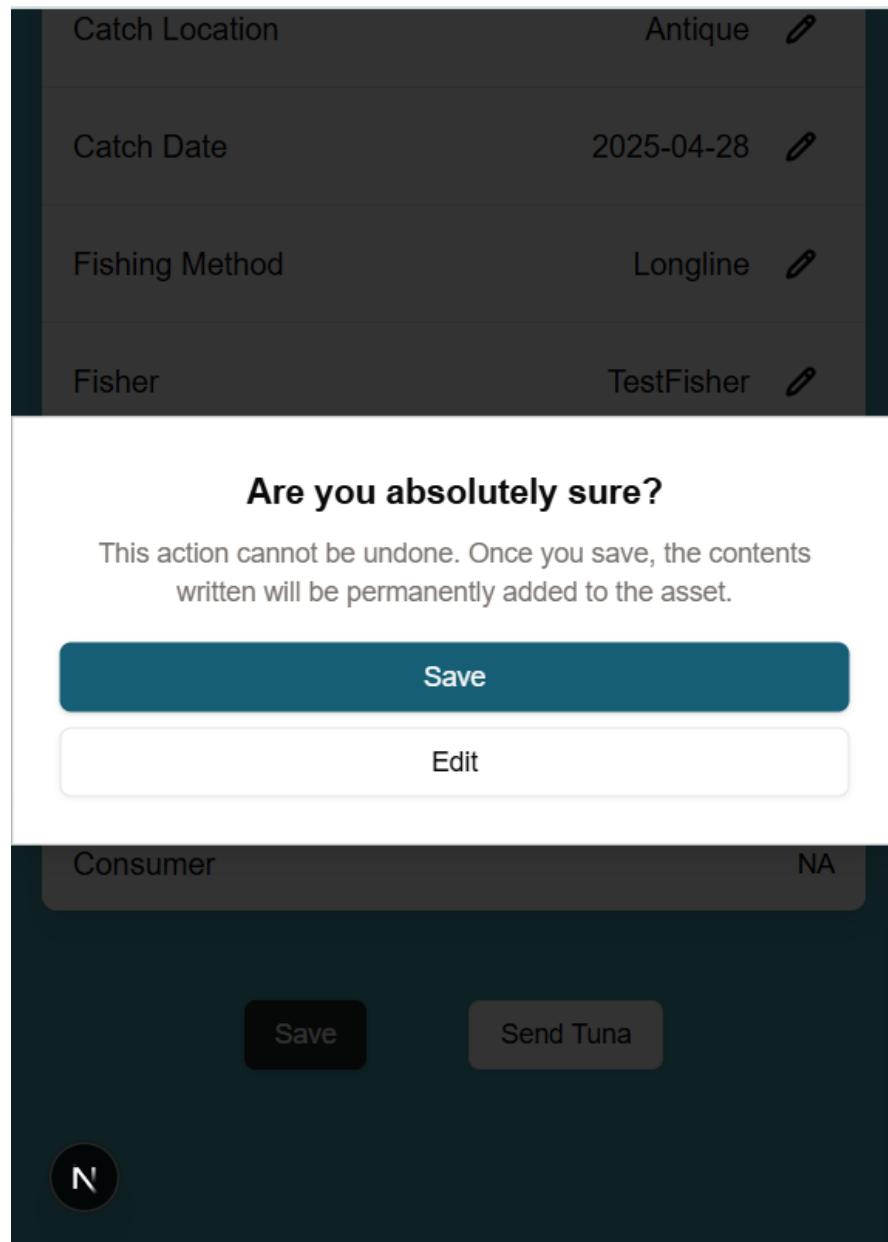


Figure 4.23: Save Details

|                   |            |                                                                                     |
|-------------------|------------|-------------------------------------------------------------------------------------|
| 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

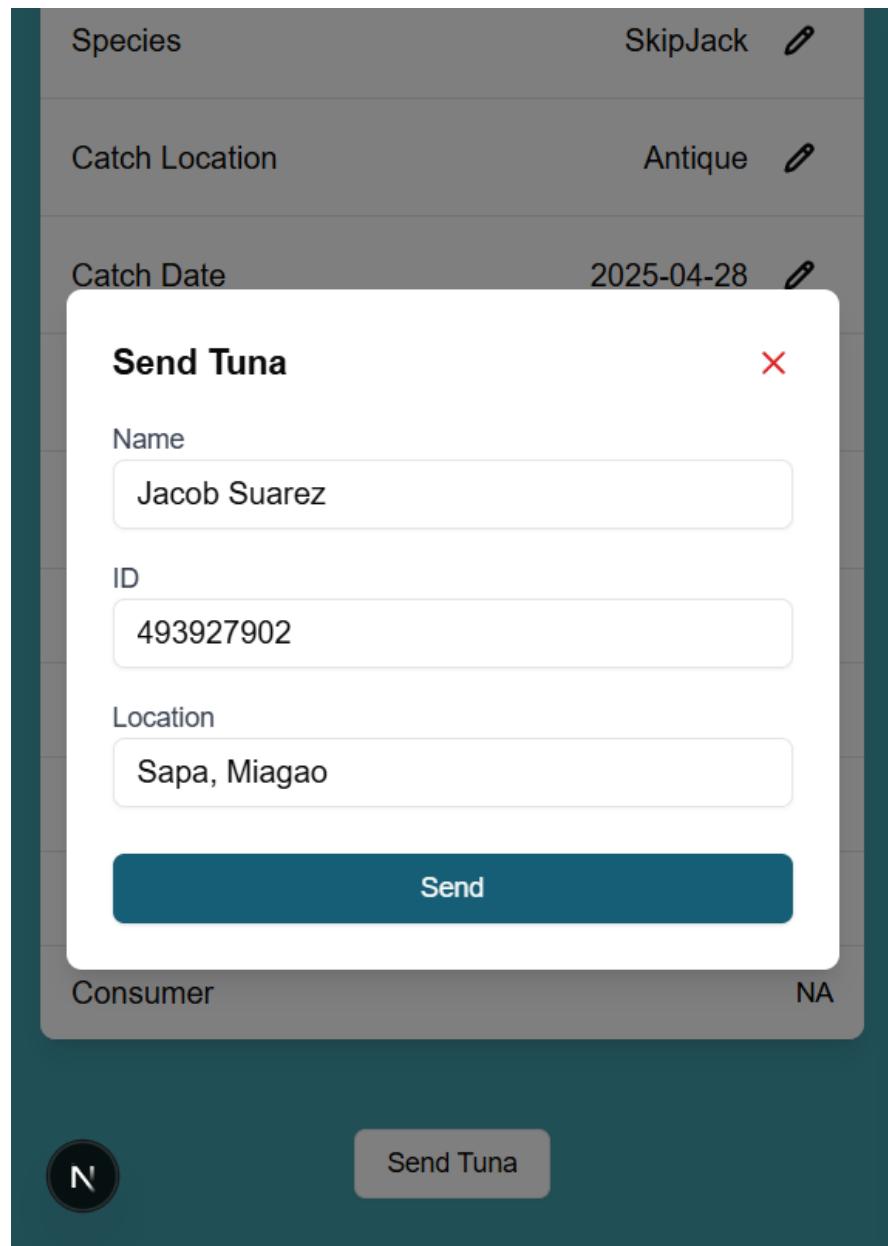


Figure 4.25: Send Asset

<sup>1325</sup> **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.

<sup>1326</sup> When taken as a whole, the respondents have average feedback in asset tracking  
<sup>1327</sup> but when classified by stakeholder, the fishermen ( $M = 4.0$ ) and consumers ( $M$   
<sup>1328</sup>  $= 4.0$ ) had good feedback in tracking , while the supplier and retailers have an  
<sup>1329</sup> average rating ( $M = 3.0$ ). For verifying tuna assets, the entire group has an average  
<sup>1330</sup> feedback. When classified by stakeholder, the fishermen ( $M = 3.33$ ) and consumers  
<sup>1331</sup> ( $M = 3.67$ ) have average ratings. For real-time updates, the respondents, when  
<sup>1332</sup> taken as a whole, have an average feedback. When classified by stakeholder, the  
<sup>1333</sup> fishermen ( $M = 3.78$ ) have an average rating, while both supplier and retailers ( $M$   
<sup>1334</sup>  $= 4.0$ ) and consumers ( $M = 4.0$ ) have good ratings. For smart contract execution,  
<sup>1335</sup> the respondents, when taken as a whole, also have an average feedback. When  
<sup>1336</sup> classified according to stakeholder, the fishermen have a fair rating ( $M = 2.33$ ),

- <sup>1337</sup> the supplier and retailers have average ratings ( $M = 3.25$ ) and the consumers have  
<sup>1338</sup> 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.

- <sup>1339</sup> The respondents, when taken as a whole, had an average feedback in transparency.  
<sup>1340</sup> When classified by stakeholder, The fishermen have fair ratings ( $M = 2.67$ ), while  
<sup>1341</sup> both supplier and retailers ( $M = 4.0$ ) and consumers ( $M = 4.0$ ) have good ratings.  
<sup>1342</sup> In evaluating the seamless interaction of the app, the entire group has an average  
<sup>1343</sup> feedback ( $M = 3.77$ ). When classified by stakeholder, the fishermen ( $M = 1.33$ )  
<sup>1344</sup> have poor feedback, the supplier and retailers have average feedback ( $M = 3.0$ )  
<sup>1345</sup> 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.

<sup>1346</sup> As a whole, the respondents have an average feedback on efficient transactions ( $M$   
<sup>1347</sup>  $= 3.81$ ). If evaluated per stakeholder, both fishermen ( $M = 3.67$ ), supplier and  
<sup>1348</sup> retailers ( $M = 3.75$ ) evaluated average while consumers had good feedback ( $M =$   
<sup>1349</sup>  $4.0$ ). For data security, the entire group has an average feedback ( $M = 3.31$ ). The  
<sup>1350</sup> fishermen have fair evaluation ( $M = 2.67$ ), supplier and retailers ( $M = 3.25$ ) have  
<sup>1351</sup> 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.

<sup>1352</sup> It shows the frequency of intuitive interface among the respondents when taken  
<sup>1353</sup> as a whole is average ( $M = 3.83$ ). When classified according to stakeholder,  
<sup>1354</sup> both fishermen ( $M = 4.0$ ) and consumers ( $M = 4.0$ ) have good ratings, while the  
<sup>1355</sup> supplier and retailers ( $M = 3.5$ ) have average ratings. For cross-platform usage,  
<sup>1356</sup> the entire group rated good ( $M = 4.14$ ). When classified according to stakeholder,  
<sup>1357</sup> both fishermen ( $M = 4.0$ ) and consumers ( $M = 4.1$ ) also have good ratings, while  
<sup>1358</sup> supplier and retailers ( $M = 3.75$ ) have average. For visual clarity, the entire group  
<sup>1359</sup> rated average ( $M = 3.80$ ). When classified by each stakeholder, both fishermen  
<sup>1360</sup> ( $M = 3.33$ ) and supplier and retailers ( $M = 3.75$ ) have average ratings, while

<sub>1361</sub> consumers ( $M = 4.33$ ) have good ratings.

| <b>Ease of Use</b>               | <b>Stakeholder</b>     | <b>Mean</b> | <b>Description</b> |
|----------------------------------|------------------------|-------------|--------------------|
| 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.

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

| <b>Feasibility</b>                  | <b>Stakeholder</b>     | <b>Mean</b> | <b>Description</b> |
|-------------------------------------|------------------------|-------------|--------------------|
| 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.

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

1371 4.0) and supplier and retailers ( $M = 4.5$ ) have good feedback in system integration,  
1372 while the consumers( $M = 3.67$ ) have an average rating. The frequency of consumer  
1373 use among stakeholders, when taken as a whole, have good feedback ( $M = 4.03$ ).  
1374 When analyzed individually, both the fishermen ( $M = 4.0$ ) and consumers ( $M =$   
1375 4.33) have a good rating, while an average rating for the supplier and retailers ( $M$   
1376 = 3.75).

### 1377 4.7.3 Results and Analysis

1378 For functionality, fishermen rated asset tracking positively but identified chal-  
1379 lenges in real-time updates and smart contract execution. Suppliers and retail-  
1380 ers similarly found strengths in verification and real-time updates, although as-  
1381 set tracking averaged slightly lower. Consumers consistently rated functionality  
1382 highly, particularly appreciating the system's ability to support smart contracts.

1383 Regarding end-user needs, fishermen provided the lowest ratings, highlighting  
1384 difficulties with seamless interaction, considering the digital barrier that they ex-  
1385 perience while interacting with the app. Suppliers, retailers, and consumers gave  
1386 higher scores, suggesting that while transparency was generally achieved, the sys-  
1387 tem's ease of interaction required further improvement for all users.

1388 In terms of performance, fishermen and suppliers reported average transaction  
1389 efficiency but noted concerns regarding data integrity and security. Consumers,  
1390 on the other hand, expressed confidence in both areas.

1391 For usability, consumers provided the highest ratings, emphasizing the intuitive  
1392 design and accessibility of the interface. Fishermen and suppliers also rated us-

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

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

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



# <sup>1404</sup> Chapter 5

## <sup>1405</sup> Conclusion

### <sup>1406</sup> 5.1 Overview

<sup>1407</sup> This chapter presents the key findings and conclusion of the study. It also discusses  
<sup>1408</sup> how the system addresses the challenges of traceability within the tuna supply  
<sup>1409</sup> chain. This chapter also provides recommendations in enhancing the system's  
<sup>1410</sup> functionality and usability, ensuring that the SeaXChange app continues to meet  
<sup>1411</sup> evolving needs of its users.

### <sup>1412</sup> 5.2 Conclusion

<sup>1413</sup> This study aimed to develop and evaluate SeaXChange, which is a blockchain-  
<sup>1414</sup> driven app designed to enhance transparency, traceability and accountability  
<sup>1415</sup> within the tuna supply chain. Through the adaption of Scrum, the team was

<sup>1416</sup> able to develop a functional prototype that was based on iterative development  
<sup>1417</sup> to achieve goals.

<sup>1418</sup> The results from the gathered data suggests that the app has effectively addressed  
<sup>1419</sup> key challenges in traceability and accountability, especially among suppliers and  
<sup>1420</sup> consumers. The stakeholders consistently rated the system as good. However,  
<sup>1421</sup> some areas need improvement, especially in ensuring seamless interaction and  
<sup>1422</sup> data security for fishermen.

<sup>1423</sup> Overall, SeaXChange demonstrates strong potential as a technological solution for  
<sup>1424</sup> promoting transparency, combating illegal, unreported, and unregulated (IUU)  
<sup>1425</sup> fishing, and empowering stakeholders across the tuna supply chain.

### <sup>1426</sup> 5.3 Recommendations

<sup>1427</sup> After analyzing and interpreting the gathered data, the researchers had identified  
<sup>1428</sup> some improvements and recommendations for the further development and  
<sup>1429</sup> implementation of the SeaXChange app.

#### <sup>1430</sup> 1. Incorporation of Local Language

<sup>1431</sup> Since most of the target users are within Miagao, the system could provide  
<sup>1432</sup> multi-language support, including the Karay-a language. This will improve  
<sup>1433</sup> guidance through clearer button descriptions.

#### <sup>1434</sup> 2. Utilization of IoT

<sup>1435</sup> The system could use Internet of Things (IoT) in verifying the fisherman's  
<sup>1436</sup> location. This will add more accountability in tracing the fisherman's cur-

<sup>1437</sup> rent location.

<sup>1438</sup> **3. Inclusion of User Manual**

<sup>1439</sup> To further enhance the experience of its users, the researcher could provide  
<sup>1440</sup> printed or digital copies on the system's functionalities. This will help users  
<sup>1441</sup> navigate through the system without being lost.

<sup>1442</sup> **5.4 Summary**

<sup>1443</sup> In conclusion, the development of the SeaXChange app highlights the critical role  
<sup>1444</sup> of emerging technologies in providing solution regarding the traceability, trans-  
<sup>1445</sup> parency and accountability within the tuna supply chain. While the system has  
<sup>1446</sup> demonstrated strong potential, continuous improvements are still needed to en-  
<sup>1447</sup> sure its effectiveness. Moreover, further development and usability enhancements  
<sup>1448</sup> will be essential in attaining SeaXChange's goal of creating a more traceable,  
<sup>1449</sup> transparent and accountable tuna industry.



<sup>1450</sup>

# Chapter 6

<sup>1451</sup>

## References

- <sup>1452</sup> Ahamed, N. N., Karthikeyan, P., Anandaraj, S., & Vignesh, R. (2020). *Sea food supply chain management using blockchain.* <https://ieeexplore.ieee.org/abstract/document/9074473>.
- <sup>1453</sup>
- <sup>1454</sup>
- <sup>1455</sup> 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>.
- <sup>1456</sup>
- <sup>1457</sup>
- <sup>1458</sup> 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
- <sup>1459</sup>
- <sup>1460</sup>
- <sup>1461</sup>
- <sup>1462</sup> 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>.
- <sup>1463</sup>
- <sup>1464</sup>
- <sup>1465</sup>
- <sup>1466</sup> Asche, F., Garlock, T. M., Anderson, J. L., Bush, S. R., Smith, M. D., Anderson,

- 1467 C. M., ... Vannuccini, S. (2018). *Three pillars of sustainability in fisheries*.  
1468 <https://www.pnas.org/doi/abs/10.1073/pnas.1807677115>.
- 1469 Cocco, L., & Mannaro, K. (2021). *Blockchain in agri-food traceability systems: 1470 a model proposal for a typical italian food product*. [https://ieeexplore.iee.org/abstract/document/9425984](https://ieeexplore.1471 ieee.org/abstract/document/9425984).
- 1472 Cole, R., Stevenson, M., & Aitken, J. (2019). *Blockchain technology: implications 1473 for operations and supply chain management*. [https://www.emerald.com/insight/content/doi/10.1108/SCM-09-2018-0309/full/html#sec006](https://www.emerald.com/1474 insight/content/doi/10.1108/SCM-09-2018-0309/full/html#sec006).
- 1475 Cordova, R. S., Maata, R. L. R., Epoc, F. J., & Alshar'e, M. (2021). *Challenges 1476 and opportunities of using blockchain in supply chain management*. [http://www.gbmrrjournal.com/pdf/v13n3/V13N3-18.pdf](http://1477 www.gbmrrjournal.com/pdf/v13n3/V13N3-18.pdf).
- 1478 CSCMP. (2024). *Cscmp supply chain management definitions and glossary*. [https://cscmp.org/CSCMP/CSCMP/Educate/SCM\\_Definitions\\_and\\_1480 Glossary\\_of\\_Terms.aspx](https://cscmp.org/CSCMP/CSCMP/Educate/SCM_Definitions_and_1479 Glossary_of_Terms.aspx).
- 1481 Delfino, A. N. (2023). *Analysing the value chain of skipjack tuna(katsuwonus 1482 pelamis) in partido district, camarinessur, philippines*. [https://www.underscienceonline.com/doi/epdf/10.1504/IJVCM.2023.129271](https://www.1483 inderscienceonline.com/doi/epdf/10.1504/IJVCM.2023.129271).
- 1484 Digal, L. N., Placencia, S. G. P., & Balgos, C. Q. (2017). *Market assessment on the 1485 incentives and disincentives for the adoption of sustainable practices along 1486 the tuna value chain in region 12, philippines*. [https://www.sciencedirect.com/science/article/abs/pii/S0308597X17301197](https://www.sciencedirect.1487 com/science/article/abs/pii/S0308597X17301197).
- 1488 Grantham, A., Pandan, M. R., Roxas, S., & Hitchcock, B. (2022). *Overcoming 1489 catch data collection challenges and traceability implementation barriers in 1490 a sustainable, small-scale fishery*. [https://www.mdpi.com/2071-1050/14/3/1179](https://www.mdpi.com/2071-1050/14/1491 3/1179).
- 1492 Hackius, N., & Petersen, M. (2017). *Blockchain in logistics and supply chain: Trick*

- 1493        or treat?        <https://www.researchgate.net/publication/318724655>  
 1494        \_Blockchain\_in\_Logistics\_and\_Supply\_Chain\_Trick\_or\_Treat.
- 1495    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](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)
- 1500    Hyperledger Foundation. (2024). Hyperledger fabric documentation [Computer software manual]. Retrieved from <https://hyperledger-fabric.readthedocs.io/> (Available at <https://hyperledger-fabric.readthedocs.io/>)
- 1504    Islam, S., & Cullen, J. M. (2021). *Food traceability: A generic theoretical framework*. <https://doi.org/10.1016/j.foodcont.2020.107848>.
- 1507    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>.
- 1510    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>.
- 1513    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>.
- 1516    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](https://www.eria.org/uploads/media/RURB_2018_FullReport.pdf#page=221).

- 1519 Macusi, E. D., Castro, M. M. C., Nallos, I. M., & Perales, C. P. (2023).  
1520 *Fishers' communication as a critical factor for tuna catches and potential*  
1521 *benefits of traceability draws small-scale fishers to program.* <https://www.sciencedirect.com/science/article/pii/S0964569123003873>.
- 1522
- 1523 Macusi, E. D., da Costa-Neves, A. C., Tipudan, C. D., & Babaran, R. P. (2023).  
1524 *Closed season and the distribution of small-scale fisheries fishing effort in*  
1525 *davao gulf, philippines.* <https://www.mdpi.com/2673-4060/4/1/4>.
- 1526 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>.
- 1527
- 1528
- 1529
- 1530 Mullon, C., Guillotreau, P., Galbraith, E. D., Fortilus, J., Chaboud, C., Bopp, L.,  
1531 ... Kaplan, D. M. (2017). *Exploring future scenarios for the global supply*  
1532 *chain of tuna.* <https://doi.org/10.1016/j.dsr2.2016.08.004>.
- 1533 Nassar, M., Rottenstreich, O., & Orda, A. (2024, February). Cfto:  
1534 Communication-aware fairness in blockchain transaction ordering. *IEEE*  
1535 *Transactions on Network and Service Management*, 21(1), 490–506. doi:  
1536 10.1109/TNSM.2023.3298201
- 1537 Nepomuceno, L. T., Bacordo, R. S., Camu, D. G. Y., & Ramiscal, R. V. (2020).  
1538 *Abundance, distribution, and diversity of tuna larvae (family scombridae)*  
1539 *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).
- 1540
- 1541
- 1542
- 1543 Obura, D. O., Aeby, G., Amornthammarong, N., Appeltans, W., Bax,  
1544 N., Bishop, J., ... Wongbusarakum, S. (2019). *Coral reef*

- 1545 monitoring, reef assessment technologies, and ecosystem-based man-  
1546 agement. [https://www.frontiersin.org/journals/marine-science/  
1547 articles/10.3389/fmars.2019.00580/full](https://www.frontiersin.org/journals/marine-science/articles/10.3389/fmars.2019.00580/full).
- 1548 Pacoma, A. U., & Yap-Dejeto, L. G. (2019). *Health risk assessment: Total  
1549 mercury in canned tuna and in yellowfin and frigate tuna caught from leyte  
1550 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](https://www.researchgate.net/publication/<br/>1551 340827869_Health_Risk_Assessment_Total_Mercury_in_Canned_Tuna<br/>1552 _and_in_Yellowfin_and_Frigate_Tuna_Caught_from_Leyte_Gulf_and<br/>1553 _Philippine_Sea).
- 1554 Paillin, D., Tupan, J., Paillin, J., Latuny, W., & Lawalata, V. (2022).  
1555 *Risk assessment and risk mitigation in a sustainable tuna sup-  
1556 ply chain.* [https://www.actalogistica.eu/issues/2022/I\\_2022\\_06  
\\_Paillin\\_Tupan\\_Paillin\\_Latuny\\_Lawalata.pdf](https://www.actalogistica.eu/issues/2022/I_2022_06<br/>1557 _Paillin_Tupan_Paillin_Latuny_Lawalata.pdf).
- 1558 Parenrenge, S. M., Pujawan, N., Karningsih, P. D., & Engelseth, P. (2016).  
1559 *Mitigating risk in the tuna supply through traceability system develop-  
1560 ment.* [https://himolde.brage.unit.no/himolde-xmlui/bitstream/  
handle/11250/3052893/Cristin-id\\_1318891\\_engelseth.pdf?sequence=  
1&isAllowed=y](https://himolde.brage.unit.no/himolde-xmlui/bitstream/<br/>1561 handle/11250/3052893/Cristin-id_1318891_engelseth.pdf?sequence=<br/>1562 1&isAllowed=y).
- 1563 Peruzy, M. F., Murru, N., Perugini, A. G., Capuano, F., Delibato, E., Mer-  
1564 cogliano, R., ... Proroga, Y. T. R. (2017). *Evaluation of virulence genes  
1565 in yersinia enterocolitica strains using sybr green real-time pcr.* [https://www.sciencedirect.com/science/article/pii/S0740002016304555](https://<br/>1566 www.sciencedirect.com/science/article/pii/S0740002016304555).
- 1567 Pomeroy, R., & Courtney, C. A. (2018). *The philippines context for marine  
1568 tenure and small-scale fisheries.* [https://sci-hub.st/https://doi.org/  
10.1016/j.marpol.2018.05.030](https://sci-hub.st/https://doi.org/<br/>1569 10.1016/j.marpol.2018.05.030).
- 1570 Sarmah, S. S. (2018). *Understanding blockchain technology.* <http://article>

- 1571 .sapub.org/10.5923.j.computer.20180802.02.html.
- 1572 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>.
- 1573
- 1574
- 1575 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>.
- 1576
- 1577
- 1578
- 1579 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](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)
- 1580
- 1581
- 1582
- 1583
- 1584 Tiwari, U. (2020). *Application of blockchain in agri-food supply chain*. <https://doi.org/10.33258/bioex.v2i2.233>.
- 1585
- 1586 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>.
- 1587
- 1588
- 1589
- 1590 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
- 1591
- 1592
- 1593
- 1594 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>.
- 1595
- 1596

1597    Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). *An overview of*  
1598    *blockchain technology: Architecture, consensus, and future trends.* [https://  
1599    ieeexplore.ieee.org/abstract/document/8029379.](https://ieeexplore.ieee.org/abstract/document/8029379)



1600 **Appendix A**

1601 **Code Snippets**

```
1602
1603 const checkAssetAccess = (
1604 asset, userIdentifier, role
1605) => {
1606
1607 switch (role.toLowerCase()) {
1608
1609 case 'fisher':
1610 if (asset.Fisher === userIdentifier)
1611 return { hasAccess: true, accessType
1612 : 'full' };
1613
1614 break;
1615
1616 case 'supplier':
1617 if (asset.Supplier ===
1618 userIdentifier)
1619 return { hasAccess: true, accessType
1620 : 'full' };
1621
1622 break;
1623
1624 case 'retailer':
```

```

1619: if (asset.Retailers?.includes(
1620: userIdentifier))
1621:
1622: return { hasAccess: true, accessType
1623: : 'full' };
1624:
1625: break;
1626:
1627: case 'consumer':
1628:
1629: if (asset.Consumers?.includes(
1630: userIdentifier))
1631:
1632: return { hasAccess: true, accessType
1633: : 'full' };
1634:
1635: break;
1636:
1637: }
1638:
1639: if (role.toLowerCase() === 'consumer') {
1640:
1641: return { hasAccess: true, accessType
1642: : 'readonly' };
1643:
1644: }
1645:
1646: return { hasAccess: false, accessType: 'none'
1647:
1648: };
1649:
1650: };

```

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

```

1641:
1642: const grpc = require('@grpc/grpc-js');
1643:
1644: const { connect, hash, signers } = require(
1645: '@hyperledger/fabric-gateway');

```

```

1646i const fs = require('node:fs/promises');
1647i
1648i const path = require('node:path');
1649i
1650i const channelName = envOrDefault('CHANNEL_NAME', 'mychannel');
1651i
1652i const chaincodeName = envOrDefault('CHAINCODE_NAME', 'basic');
1653i
1654i const mspId = envOrDefault('MSP_ID', 'Org1MSP');
1655i
1656i // Path to crypto materials.
1657i
1658i const cryptoPath = envOrDefault(
1659i 'CRYPTO_PATH',
1660i path.resolve(
1661i __dirname,
1662i '..',
1663i '..',
1664i 'test-network',
1665i 'organizations',
1666i 'peerOrganizations',
1667i 'org1.example.com'
1668i)
1669i);
1670i
1671i const keyDirectoryPath = envOrDefault(
1672i 'KEY_DIRECTORY_PATH',

```

```
1673) path.resolve(
1674) cryptoPath,
1675) 'users',
1676) 'User1@org1.example.com',
1677) 'msp',
1678) 'keystore'
1679))
1680);
1681;
1682) const certDirectoryPath = envOrDefault(
1683) 'CERT_DIRECTORY_PATH',
1684) path.resolve(
1685) cryptoPath,
1686) 'users',
1687) 'User1@org1.example.com',
1688) 'msp',
1689) 'signcerts'
1690))
1691);
1692;
1693) const tlsCertPath = envOrDefault(
1694) 'TLS_CERT_PATH',
1695) path.resolve(cryptoPath, 'peers', 'peer0.org1.
1696) example.com', 'tls', 'ca.crt')
1697));
1698;
1699) const peerEndpoint = envOrDefault('PEER_ENDPOINT', '
```

```

1700 localhost:7051');
1701
1702 const peerHostAlias = envOrDefault('PEER_HOST_ALIAS',
1703 'peer0.org1.example.com');
1704
1705 const utf8Decoder = new TextDecoder();
1706
1707 const assetId = asset${String(Date.now())};

```

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

```

1707
1708 type Asset struct {
1709
1710 ID
1711 string `json:"ID"`
1712
1713 Species
1714 string `json:"Species"`
1715
1716 Weight
1717 float64 `json:"Weight"`
1718
1719 CatchLocation
1720 string `json:"CatchLocation"`
1721
1722 CatchDate
1723 string `json:"CatchDate"`
1724
1725 FishingMethod
1726 string `json:"FishingMethod"`
1727
1728 Fisher
1729 string `json:"Fisher"`
1730
1731 Supplier
1732 string `json:"Supplier"`
1733
1734 SellingLocationSupplier
1735 string `json:"SellingLocationSupplier"`

```

```

1727: Retailers
1728 [] string `json:"Retailers"
1729 " "
1730: SellingLocationRetailers []
1731 string `json:"SellingLocationRetailers"
1732: Consumers
1733 [] string `json:"Consumers"
1734 " "
1735: }
1736

```

Listing A.3: Asset Data Structure

```

1737
1738: func (s *SmartContract) CreateAsset(ctx contractapi.
1739
1740 TransactionContextInterface, id string, species
1741
1742 string, weight float64, catchLocation string,
1743
1744 catchDate string, fishingMethod string, fisher
1745
1746 string) error {
1747
1748 exists, err := s.AssetExists(ctx, id)
1749
1750 if err != nil {
1751
1752 return err
1753 }
1754
1755 if exists {
1756
1757 return fmt.Errorf("the asset %s
1758
1759 already exists", id)
1760 }
1761
1762 asset := Asset{
1763
1764 ID:
1765
1766 id,
1767
1768 Species:
1769
1770 species,
1771
1772 Weight:
1773
1774 weight,
1775
1776 CatchLocation:
1777
1778 catchLocation,
1779
1780 CatchDate:
1781
1782 catchDate,
1783
1784 FishingMethod:
1785
1786 fishingMethod,
1787
1788 Fisher:
1789
1790 fisher,
1791
1792 Retailers:
1793
1794 [] string `json:"Retailers"
1795 " "
1796 }
1797
1798 err = s.CreateAsset(ctx, asset)
1799
1800 if err != nil {
1801
1802 return err
1803 }
1804
1805 return nil
1806
1807 }
1808
1809 return nil
1810
1811 }
1812
1813 return nil
1814
1815 }
1816
1817 return nil
1818
1819 }
1820
1821 return nil
1822
1823 }
1824
1825 return nil
1826
1827 }
1828
1829 return nil
1830
1831 }
1832
1833 return nil
1834
1835 }
1836
1837 return nil
1838
1839 }
1840
1841 return nil
1842
1843 }
1844
1845 return nil
1846
1847 }
1848
1849 return nil
1850
1851 }
1852
1853 return nil
1854
1855 }
1856
1857 return nil
1858
1859 }
1860
1861 return nil
1862
1863 }
1864
1865 return nil
1866
1867 }
1868
1869 return nil
1870
1871 }
1872
1873 return nil
1874
1875 }
1876
1877 return nil
1878
1879 }
1880
1881 return nil
1882
1883 }
1884
1885 return nil
1886
1887 }
1888
1889 return nil
1890
1891 }
1892
1893 return nil
1894
1895 }
1896
1897 return nil
1898
1899 }
1900
1901 return nil
1902
1903 }
1904
1905 return nil
1906
1907 }
1908
1909 return nil
1910
1911 }
1912
1913 return nil
1914
1915 }
1916
1917 return nil
1918
1919 }
1920
1921 return nil
1922
1923 }
1924
1925 return nil
1926
1927 }
1928
1929 return nil
1930
1931 }
1932
1933 return nil
1934
1935 }
1936
1937 return nil
1938
1939 }
1940
1941 return nil
1942
1943 }
1944
1945 return nil
1946
1947 }
1948
1949 return nil
1950
1951 }
1952
1953 return nil
1954
1955 }
1956
1957 return nil
1958
1959 }
1960
1961 return nil
1962
1963 }
1964
1965 return nil
1966
1967 }
1968
1969 return nil
1970
1971 }
1972
1973 return nil
1974
1975 }
1976
1977 return nil
1978
1979 }
1980
1981 return nil
1982
1983 }
1984
1985 return nil
1986
1987 }
1988
1989 return nil
1990
1991 }
1992
1993 return nil
1994
1995 }
1996
1997 return nil
1998
1999 }
1999
2000 return nil
2001
2002 }
2003
2004 return nil
2005
2006 }
2007
2008 return nil
2009
2010 }
2011
2012 return nil
2013
2014 }
2015
2016 return nil
2017
2018 }
2019
2020 return nil
2021
2022 }
2023
2024 return nil
2025
2026 }
2027
2028 return nil
2029
2030 }
2031
2032 return nil
2033
2034 }
2035
2036 return nil
2037
2038 }
2039
2040 return nil
2041
2042 }
2043
2044 return nil
2045
2046 }
2047
2048 return nil
2049
2050 }
2051
2052 return nil
2053
2054 }
2055
2056 return nil
2057
2058 }
2059
2060 return nil
2061
2062 }
2063
2064 return nil
2065
2066 }
2067
2068 return nil
2069
2070 }
2071
2072 return nil
2073
2074 }
2075
2076 return nil
2077
2078 }
2079
2080 return nil
2081
2082 }
2083
2084 return nil
2085
2086 }
2087
2088 return nil
2089
2090 }
2091
2092 return nil
2093
2094 }
2095
2096 return nil
2097
2098 }
2099
2099
2100 return nil
2101
2102 }
2103
2104 return nil
2105
2106 }
2107
2108 return nil
2109
2110 }
2111
2112 return nil
2113
2114 }
2115
2116 return nil
2117
2118 }
2119
2120 return nil
2121
2122 }
2123
2124 return nil
2125
2126 }
2127
2128 return nil
2129
2130 }
2131
2132 return nil
2133
2134 }
2135
2136 return nil
2137
2138 }
2139
2140 return nil
2141
2142 }
2143
2144 return nil
2145
2146 }
2147
2148 return nil
2149
2150 }
2151
2152 return nil
2153
2154 }
2155
2156 return nil
2157
2158 }
2159
2160 return nil
2161
2162 }
2163
2164 return nil
2165
2166 }
2167
2168 return nil
2169
2170 }
2171
2172 return nil
2173
2174 }
2175
2176 return nil
2177
2178 }
2179
2180 return nil
2181
2182 }
2183
2184 return nil
2185
2186 }
2187
2188 return nil
2189
2190 }
2191
2192 return nil
2193
2194 }
2195
2196 return nil
2197
2198 }
2199
2199
2200 return nil
2201
2202 }
2203
2204 return nil
2205
2206 }
2207
2208 return nil
2209
2210 }
2211
2212 return nil
2213
2214 }
2215
2216 return nil
2217
2218 }
2219
2219
2220 return nil
2221
2222 }
2223
2224 return nil
2225
2226 }
2227
2228 return nil
2229
2230 }
2231
2232 return nil
2233
2234 }
2235
2236 return nil
2237
2238 }
2239
2239
2240 return nil
2241
2242 }
2243
2244 return nil
2245
2246 }
2247
2248 return nil
2249
2250 }
2251
2252 return nil
2253
2254 }
2255
2256 return nil
2257
2258 }
2259
2259
2260 return nil
2261
2262 }
2263
2264 return nil
2265
2266 }
2267
2268 return nil
2269
2270 }
2271
2272 return nil
2273
2274 }
2275
2276 return nil
2277
2278 }
2279
2279
2280 return nil
2281
2282 }
2283
2284 return nil
2285
2286 }
2287
2288 return nil
2289
2290 }
2291
2292 return nil
2293
2294 }
2295
2296 return nil
2297
2298 }
2299
2299
2300 return nil
2301
2302 }
2303
2304 return nil
2305
2306 }
2307
2308 return nil
2309
2310 }
2311
2312 return nil
2313
2314 }
2315
2316 return nil
2317
2318 }
2319
2319
2320 return nil
2321
2322 }
2323
2324 return nil
2325
2326 }
2327
2328 return nil
2329
2330 }
2331
2332 return nil
2333
2334 }
2335
2336 return nil
2337
2338 }
2339
2339
2340 return nil
2341
2342 }
2343
2344 return nil
2345
2346 }
2347
2348 return nil
2349
2350 }
2351
2352 return nil
2353
2354 }
2355
2356 return nil
2357
2358 }
2359
2359
2360 return nil
2361
2362 }
2363
2364 return nil
2365
2366 }
2367
2368 return nil
2369
2370 }
2371
2372 return nil
2373
2374 }
2375
2376 return nil
2377
2378 }
2379
2379
2380 return nil
2381
2382 }
2383
2384 return nil
2385
2386 }
2387
2388 return nil
2389
2390 }
2391
2392 return nil
2393
2394 }
2395
2396 return nil
2397
2398 }
2399
2399
2400 return nil
2401
2402 }
2403
2404 return nil
2405
2406 }
2407
2408 return nil
2409
2410 }
2411
2412 return nil
2413
2414 }
2415
2416 return nil
2417
2418 }
2419
2419
2420 return nil
2421
2422 }
2423
2424 return nil
2425
2426 }
2427
2428 return nil
2429
2430 }
2431
2432 return nil
2433
2434 }
2435
2436 return nil
2437
2438 }
2439
2439
2440 return nil
2441
2442 }
2443
2444 return nil
2445
2446 }
2447
2448 return nil
2449
2450 }
2451
2452 return nil
2453
2454 }
2455
2456 return nil
2457
2458 }
2459
2459
2460 return nil
2461
2462 }
2463
2464 return nil
2465
2466 }
2467
2468 return nil
2469
2470 }
2471
2472 return nil
2473
2474 }
2475
2476 return nil
2477
2478 }
2479
2479
2480 return nil
2481
2482 }
2483
2484 return nil
2485
2486 }
2487
2488 return nil
2489
2490 }
2491
2492 return nil
2493
2494 }
2495
2496 return nil
2497
2498 }
2499
2499
2500 return nil
2501
2502 }
2503
2504 return nil
2505
2506 }
2507
2508 return nil
2509
2510 }
2511
2512 return nil
2513
2514 }
2515
2516 return nil
2517
2518 }
2519
2519
2520 return nil
2521
2522 }
2523
2524 return nil
2525
2526 }
2527
2528 return nil
2529
2530 }
2531
2532 return nil
2533
2534 }
2535
2536 return nil
2537
2538 }
2539
2539
2540 return nil
2541
2542 }
2543
2544 return nil
2545
2546 }
2547
2548 return nil
2549
2550 }
2551
2552 return nil
2553
2554 }
2555
2556 return nil
2557
2558 }
2559
2559
2560 return nil
2561
2562 }
2563
2564 return nil
2565
2566 }
2567
2568 return nil
2569
2570 }
2571
2572 return nil
2573
2574 }
2575
2576 return nil
2577
2578 }
2579
2579
2580 return nil
2581
2582 }
2583
2584 return nil
2585
2586 }
2587
2588 return nil
2589
2590 }
2591
2592 return nil
2593
2594 }
2595
2596 return nil
2597
2598 }
2599
2599
2600 return nil
2601
2602 }
2603
2604 return nil
2605
2606 }
2607
2608 return nil
2609
2610 }
2611
2612 return nil
2613
2614 }
2615
2616 return nil
2617
2618 }
2619
2619
2620 return nil
2621
2622 }
2623
2624 return nil
2625
2626 }
2627
2628 return nil
2629
2630 }
2631
2632 return nil
2633
2634 }
2635
2636 return nil
2637
2638 }
2639
2639
2640 return nil
2641
2642 }
2643
2644 return nil
2645
2646 }
2647
2648 return nil
2649
2650 }
2651
2652 return nil
2653
2654 }
2655
2656 return nil
2657
2658 }
2659
2659
2660 return nil
2661
2662 }
2663
2664 return nil
2665
2666 }
2667
2668 return nil
2669
2670 }
2671
2672 return nil
2673
2674 }
2675
2676 return nil
2677
2678 }
2679
2679
2680 return nil
2681
2682 }
2683
2684 return nil
2685
2686 }
2687
2688 return nil
2689
2690 }
2691
2692 return nil
2693
2694 }
2695
2696 return nil
2697
2698 }
2699
2699
2700 return nil
2701
2702 }
2703
2704 return nil
2705
2706 }
2707
2708 return nil
2709
2710 }
2711
2712 return nil
2713
2714 }
2715
2716 return nil
2717
2718 }
2719
2719
2720 return nil
2721
2722 }
2723
2724 return nil
2725
2726 }
2727
2728 return nil
2729
2730 }
2731
2732 return nil
2733
2734 }
2735
2736 return nil
2737
2737
2738 return nil
2739
2740 }
2741
2742 return nil
2743
2744 }
2745
2746 return nil
2747
2748 }
2749
2749
2750 return nil
2751
2752 }
2753
2754 return nil
2755
2756 }
2757
2758 return nil
2759
2759
2760 return nil
2761
2762 }
2763
2764 return nil
2765
2766 }
2767
2768 return nil
2769
2770 }
2771
2772 return nil
2773
2774 }
2775
2776 return nil
2777
2778 }
2779
2779
2780 return nil
2781
2782 }
2783
2784 return nil
2785
2786 }
2787
2788 return nil
2789
2790 }
2791
2792 return nil
2793
2794 }
2795
2796 return nil
2797
2798 }
2799
2799
2800 return nil
2801
2802 }
2803
2804 return nil
2805
2806 }
2807
2808 return nil
2809
2810 }
2811
2812 return nil
2813
2814 }
2815
2816 return nil
2817
2818 }
2819
2819
2820 return nil
2821
2822 }
2823
2824 return nil
2825
2826 }
2827
2828 return nil
2829
2830 }
2831
2832 return nil
2833
2834 }
2835
2836 return nil
2837
2838 }
2839
2839
2840 return nil
2841
2842 }
2843
2844 return nil
2845
2846 }
2847
2848 return nil
2849
2850 }
2851
2852 return nil
2853
2854 }
2855
2856 return nil
2857
2858 }
2859
2859
2860 return nil
2861
2862 }
2863
2864 return nil
2865
2866 }
2867
2868 return nil
2869
2870 }
2871
2872 return nil
2873
2874 }
2875
2876 return nil
2877
2878 }
2879
2879
2880 return nil
2881
2882 }
2883
2884 return nil
2885
2886 }
2887
2888 return nil
2889
2890 }
2891
2892 return nil
2893
2894 }
2895
2896 return nil
2897
2898 }
2899
2899
2900 return nil
2901
2902 }
2903
2904 return nil
2905
2906 }
2907
2908 return nil
2909
2910 }
2911
2912 return nil
2913
2914 }
2915
2916 return nil
2917
2918 }
2919
2919
2920 return nil
2921
2922 }
2923
2924 return nil
2925
2926 }
2927
2928 return nil
2929
2930 }
2931
2932 return nil
2933
2934 }
2935
2936 return nil
2937
2938 }
2939
2939
2940 return nil
2941
2942 }
2943
2944 return nil
2945
2946 }
2947
2948 return nil
2949
2950 }
2951
2952 return nil
2953
2954 }
2955
2956 return nil
2957
2958 }
2959
2959
2960 return nil
2961
2962 }
2963
2964 return nil
2965
2966 }
2967
2968 return nil
2969
2970 }
2971
2972 return nil
2973
2974 }
2975
2976 return nil
2977
2978 }
2979
2979
2980 return nil
2981
2982 }
2983
2984 return nil
2985
2986 }
2987
2988 return nil
2989
2990 }
2991
2992 return nil
2993
2994 }
2995
2996 return nil
2997
2998 }
2999
2999
3000 return nil
3001
3002 }
3003
3004 return nil
3005
3006 }
3007
3008 return nil
3009
3010 }
3011
3012 return nil
3013
3014 }
3015
3016 return nil
3017
3018 }
3019
3019
3020 return nil
3021
3022 }
3023
3024 return nil
3025
3026 }
3027
3028 return nil
3029
3030 }
3031
3032 return nil
3033
3034 }
3035
3036 return nil
3037
3038 }
3039
3039
3040 return nil
3041
3042 }
3043
3044 return nil
3045
3046 }
3047
3048 return nil
3049
3050 }
3051
3052 return nil
3053
3054 }
3055
3056 return nil
3057
3058 }
3059
3059
3060 return nil
3061
3062 }
3063
3064 return nil
3065
3066 }
3067
3068 return nil
3069
3070 }
3071
3072 return nil
3073
3074 }
3075
3076 return nil
3077
3078 }
3079
3079
3080 return nil
3081
3082 }
3083
3084 return nil
3085
3086 }
3087
3088 return nil
3089
3090 }
3091
3092 return nil
3093
3094 }
3095
3096 return nil
3097
3098 }
3099
3099
3100 return nil
3101
3102 }
3103
3104 return nil
3105
3106 }
3107
3108 return nil
3109
3110 }
3111
3112 return nil
3113
3114 }
3115
3116 return nil
3117
3118 }
3119
3119
3120 return nil
3121
3122 }
3123
3124 return nil
3125
3126 }
3127
3128 return nil
3129
3130 }
3131
3132 return nil
3133
3134 }
3135
3136 return nil
3137
3138 }
3139
3139
3140 return nil
3141
3142 }
3143
3144 return nil
3145
3146 }
3147
3148 return nil
3149
3150 }
3151
3152 return nil
3153
3154 }
3155
3156 return nil
3157
3158 }
3159
3159
3160 return nil
3161
3162 }
3163
3164 return nil
3165
3166 }
3167
3168 return nil
3169
3170 }
3171
3172 return nil
3173
3174 }
3175
3176 return nil
3177
3178 }
3179
3179
3180 return nil
3181
3182 }
3183
3184 return nil
3185
3186 }
3187
3188 return nil
3189
3190 }
3191
3192 return nil
3193
3194 }
3195
3196 return nil
3197
3198 }
3199
3199
3200 return nil
3201
3202 }
3203
3204 return nil
3205
3206 }
3207
3208 return nil
3209
3210 }
3211
3212 return nil
3213
3214 }
3215
3216 return nil
3217
3218 }
3219
3219
3220 return nil
3221
3222 }
3223
3224 return nil
3225
3226 }
3227
3228 return nil
3229
3230 }
3231
3232 return nil
3233
3234 }
3235
3236 return nil
3237
3238 }
3239
3239
3240 return nil
3241
3242 }
3243
3244 return nil
3245
3246 }
3247
3248 return nil
3249
3250 }
3251
3252 return nil
3253
3254 }
3255
3256 return nil
3257
3258 }
3259
3259
3260 return nil
3261
3262 }
3263
3264 return nil
3265
3266 }
3267
3268 return nil
3269
3270 }
3271
3272 return nil
3273
3274 }
3275
3276 return nil
3277
3278 }
3279
3279
3280 return nil
3281
3282 }
3283
3284 return nil
3285
3286 }
3287
3288 return nil
3289
3290 }
3291
3292 return nil
3293
3294 }
3295
3296 return nil
3297
3298 }
3299
3299
3300 return nil
3301
3302 }
3303
3304 return nil
3305
3306 }
3307
3308 return nil
3309
3310 }
3311
3312 return nil
3313
3314 }
3315
3316 return nil
3317
3318 }
3319
3319
3320 return nil
3321
3322 }
3323
3324 return nil
3325
3326 }
3327
3328 return nil
3329
3330 }
3331
3332 return nil
3333
3334 }
3335
3336 return nil
3337
3338 }
3339
3339
3340 return nil
3341
3342 }
3343
3344 return nil
3345
3346 }
3347
3348 return nil
3349
3350 }
3351
3352 return nil
3353
3354 }
3355
3356 return nil
3357
3358 }
3359
3359
3360 return nil
3361
3362 }
3363
3364 return nil
3365
3366 }
3367
3368 return nil
3369
3370 }
3371
3372 return nil
3373
3374 }
3375
3376 return nil
3377
3378 }
3379
3379
3380 return nil
3381
3382 }
3383
3384 return nil
3385
3386 }
3387
3388 return nil
3389
3390 }
3391
3392 return nil
3393
3394 }
3395
3396 return nil
3397
3398 }
3399
3399
3400 return nil
3401
3402 }
3403
3404 return nil
3405
3406 }
3407
3408 return nil
3409
3410 }
3411
3412 return nil
3413
3414 }
3415
3416 return nil
3417
3418 }
3419
3419
3420 return nil
3421
3422 }
3423
3424 return nil
3425
3426 }
3427
3428 return nil
3429
3430 }
3431
3432 return nil
3433
3434 }
3435
3436 return nil
3437
3438 }
3439
3439
3440 return nil
3441
3442 }
3443
3444 return nil
3445
3446 }
3447
3448 return nil
3449
3450 }
3451
3452 return nil
3453
3454 }
3455
3456 return nil
3457
3458 }
3459
3459
3460 return nil
3461
3462 }
3463
3464 return nil
3465
3466 }
3467
3468 return nil
3469
3470 }
3471
3472 return nil
3473
3474 }
3475
3476 return nil
3477
3478 }
3479
3479
3480 return nil
3481
3482 }
3483
3484 return nil
3485
3486 }
3487
3488 return nil
3489
3490 }
3491
3492 return nil
3493
3494 }
3495
3496 return nil
3497
3498 }
3499
3499
3500 return nil
3501
3502 }
3503
3504 return nil
3505
3506 }
3507
3508 return nil
3509
3510 }
3511
3512 return nil
3513
3514 }
3515
3516 return nil
3517
351
```

```

1754: Species: species,
1755: Weight: weight,
1756: CatchLocation: catchLocation,
1757: CatchDate: catchDate,
1758: FishingMethod: fishingMethod,
1759: Fisher: fisher,
1760: Retailers: [] string{},
1761: SellingLocationRetailers: []
1762: string{},
1763: Consumers: [] string{},
1764: }
1765: assetJSON, err := json.Marshal(asset)
1766: if err != nil {
1767: return err
1768: }
1769:
1770: return ctx.GetStub().PutState(id, assetJSON)
1771: }
1772

```

Listing A.4: CreateAsset Function

```

1773
1774: func (s *SmartContract) TransferAsset(ctx
1775: contractapi.TransactionContextInterface, id
1776: string, role string, newParticipant string,
1777: newLocation string) (string, error) {
1778: asset, err := s.ReadAsset(ctx, id)
1779: if err != nil {
1780: return "", fmt.Errorf("failed to

```

```
1781 fetch asset: %v", err)
1782 }
1783
1784 switch role{
1785 case "Supplier":
1786 oldSupplier := asset.Supplier
1787 asset.Supplier = newParticipant
1788 asset.SellingLocationSupplier =
1789 newLocation
1790 return oldSupplier, s.SaveAsset(ctx,
1791 id, asset)
1792 case "Retailer":
1793 if !contains(asset.Retailers,
1794 newParticipant){
1795 asset.Retailers = append(
1796 asset.Retailers,
1797 newParticipant)
1798 asset.
1799 SellingLocationRetailers
1800 = append(asset.
1801 SellingLocationRetailers,
1802 newLocation)
1803 }
1804 return "", s.SaveAsset(ctx, id,
1805 asset)
1806 case "Consumer":
1807 if !contains(asset.Consumers,
```

```

1808 newParticipant) {
1809 asset.Consumers = append(
1810 asset.Consumers,
1811 newParticipant)
1812 }
1813 return "", s.SaveAsset(ctx, id,
1814 asset)
1815 default:
1816 return "", fmt.Errorf("invalid role
1817 specified: %s", role)
1818 }
1819 }
1820 }
```

Listing A.5: TransferAsset Function

```

1821
1822 func (s *SmartContract) ReadAsset(ctx contractapi.
1823 TransactionContextInterface, id string) (*Asset,
1824 error) {
1825
1826 assetJSON, err := ctx.GetStub().GetState(id)
1827
1828 if err != nil {
1829
1830 return nil, fmt.Errorf("failed to
1831 read from world state: %v", err)
1832
1833 }
1834
1835 if assetJSON == nil {
1836
1837 return nil, fmt.Errorf("the asset %s
1838 does not exist", id)
1839
1840 }
1841
1842 var asset Asset
```

```

1835) err = json.Unmarshal(assetJSON, &asset)
1836) if err != nil {
1837) return nil, err
1838) }
1839) if asset.Consumers == nil {
1840) asset.Consumers = []string{}
1841) }
1842)
1843) return &asset, nil
1844)
1845}

```

Listing A.6: ReadAsset Function

```

1846
1847) func (s *SmartContract) GetAllAssets(ctx contractapi.
1848) .TransactionContextInterface) ([]*Asset, error) {
1849)
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)
1860) queryResponse, err :=
1861) resultsIterator.Next()
1862) if err != nil {

```

```

1862: }
1863:
1864: var asset Asset
1865: err = json.Unmarshal(queryResponse.
1866: Value, &asset)
1867: if err != nil {
1868: return nil, err
1869: }
1870: if asset.Consumers == nil {
1871: asset.Consumers = []string{}
1872: }
1873: assets = append(assets, &asset)
1874: }
1875:
1876: return assets, nil
1877: }
1878

```

Listing A.7: GetAllAssets Function

```

1879
1880 async function main() {
1881 displayInputParameters();
1882
1883 const client = await newGrpcConnection();
1884
1885 const gateway = connect({
1886 client,
1887 identity: await newIdentity(),
1888 signer: await newSigner(),

```

```
1889) hash: hash.sha256,
1890) evaluateOptions: () => ({ deadline:
1891) Date.now() + 5000 }),
1892) endorseOptions: () => ({ deadline:
1893) Date.now() + 15000 }),
1894) submitOptions: () => ({ deadline:
1895) Date.now() + 5000 }),
1896) commitStatusOptions: () => ({
1897) deadline: Date.now() + 60000 }),
1898));
1899)
1900) try {
1901) const network = gateway.getNetwork(
1902) channelId);
1903) const contract = network.getContract(
1904) chaincodeName);
1905)
1906) await initLedger(contract);
1907) await getAllAssets(contract);
1908) await createAsset(contract);
1909) await transferAssetAsync(contract);
1910) await readAssetByID(contract);
1911) await updateNonExistentAsset(
1912) contract);
1913) } finally {
1914) gateway.close();
1915) client.close();
```

```

1916) }
1917) }
1918)
1919) main().catch((error) => {
1920) console.error('** FAILED to run the
1921) application:', error);
1922) process.exitCode = 1;
1923) });
1924)

```

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

## 1925 A.1 Hyperledger Fabric Deployment Instructions

1926 The following steps outline the process for deploying and testing the fish supply  
 1927 chain smart contract on Hyperledger Fabric using Google Cloud Platform.

### 1928 A.1.1 Environment Setup

#### 1929 1. Open GCP and access the VM instance:

- 1930     • Navigate to Console → Compute Engine → VM instances → start →
- 1931       click SSH
- 1932     • Alternatively: Virtual Machine → start → instance → SSH

#### 1933 2. Connect to the instance:

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

```

1935 gcloud compute ssh instance
1936 -20250322-102900 --zone=us-central1-c
1937

```

1938     3. Navigate to the test network directory:

Listing A.10: Navigate to Compose Directory

```

1939
1940 cd ~/fabric-samples/test-network/
1941
1942 compose

```

1943   A.1.2 Network Startup and Smart Contract Deployment

1944     1. Start the Hyperledger Fabric network:

Listing A.11: Start Fabric Network

```

1945
1946 sudo docker-compose -f
1947 compose-test-net.yaml
1948
1949 start

```

1950     • Deploy the chaincode:

Listing A.12: Deploy Chaincode

```

1951
1952 cd ../
1953
1954 ./network.sh deployCC -ccn
1955 basic -ccp ../asset-
1956 transfer-basic/chaincode-
1957 go -ccl go

```

1958     • Set environment path variables:

Listing A.13: Path Environment Variables

```

1959
1960 export PATH=${PWD}/../bin:
1961 $PATH
1962 export FABRIC_CFG_PATH=${PWD}
1963 /../config/
1964

```

- 1965 • Configure organization environment variables:

Listing A.14: Org1 Environment Configuration

```

1966
1967 # Environment variables for
1968 Org1
1969 export CORE_PEER_TLS_ENABLED
1970 =true
1971 export CORE_PEER_LOCALMSPID=
1972 "Org1MSP"
1973 export
1974 CORE_PEER_TLS_ROOTCERT_FILE
1975 =${PWD}/organizations/
1976 peerOrganizations/org1.
1977 example.com/peers/peer0.
1978 org1.example.com/tls/ca.
1979 crt
1980 export
1981 CORE_PEER_MSPCONFIGPATH=$
1982 ${PWD}/organizations/
1983 peerOrganizations/org1.
1984 example.com/users/
1985 Admin@org1.example.com/

```

```

1986 msp
1987 export CORE_PEER_ADDRESS=
1988 localhost:7051
1989

```

### **1990 A.1.3 Testing the Smart Contract**

#### **1991 1. Initialize the ledger:**

Listing A.15: Invoke InitLedger

```

1992
1993 peer chaincode invoke -o localhost:7050
1994 --ordererTLSHostnameOverride orderer
1995 .example.com --tls --cafile "${PWD}/
1996 organizations/ordererOrganizations/
1997 example.com/orderers/orderer.example.
1998 com/msp/tlscacerts/tlsca.example.com-
1999 cert.pem" -C mychannel -n basic --
2000 peerAddresses localhost:7051 --
2001 tlsRootCertFiles "${PWD}/
2002 organizations/peerOrganizations/org1.
2003 example.com/peers/peer0.org1.example.
2004 com/tls/ca.crt" --peerAddresses
2005 localhost:9051 --tlsRootCertFiles "${
2006 PWD}/organizations/peerOrganizations/
2007 org2.example.com/peers/peer0.org2.
2008 example.com/tls/ca.crt" -c '{"
2009 function":"InitLedger","Args":[]}'
2010

```

## 2011 2. Query assets:

Listing A.16: Query Fish Asset

```
2013 # Query a specific fish asset
2014 peer chaincode query -C mychannel -n
2015 basic -c '{"Args": ["ReadAsset", "tuna1"
2016 "]}]',
2017
2018 # Query all assets in the ledger
2019 peer chaincode query -C mychannel -n
2020 basic -c '{"Args": ["GetAllAssets"]}'
```

### 3. Shut down the network:

Listing A.17: Stop Fabric Network

```
2024 sudo docker-compose -f compose-test-net
2025 .yaml stop
2026
```

## 2027 A.1.4 Important Notes

- Ensure proper network connectivity when working with Google Cloud Platform.
  - The environment variables must be set correctly for the organization context.
  - TLS certificates are required for secure communication between nodes.
  - The commands assume the blockchain network is already provisioned on GCP.



<sup>2034</sup> **Appendix B**

<sup>2035</sup> **Resource Persons**

<sup>2036</sup> **Dr. Ricardo P. Babaran**

<sup>2037</sup> Professor of Fisheries

<sup>2038</sup> Institute of Marine Fisheries and Oceanology

<sup>2039</sup> University of the Philippines Visayas

<sup>2040</sup> rpbabaran@upv.edu.ph

<sup>2041</sup> **Engr. Noel Lucero**

<sup>2042</sup> Engineer

<sup>2043</sup> Jagnee Fishing Corp.

<sup>2044</sup> JagneeFishingCorp@outlook.com

<sup>2045</sup> **Ms. Veronica Jeruta**

<sup>2046</sup> Barangay Kagawad

<sup>2047</sup> Sapa Barangay Hall

<sup>2048</sup> veronicanave9@gmail.com



2050 **Appendix C**

2051 **Interview Request Letter**

2052 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

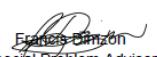
  
Francis Dimzon  
Special Problem Adviser

Figure C.1: Scanned Interview Request Letter

2053 **Appendix D**

2054 **App Demo Documentation**

2055 The photographs taken show the engangement of the interviewees during the app  
2056 demonstration.

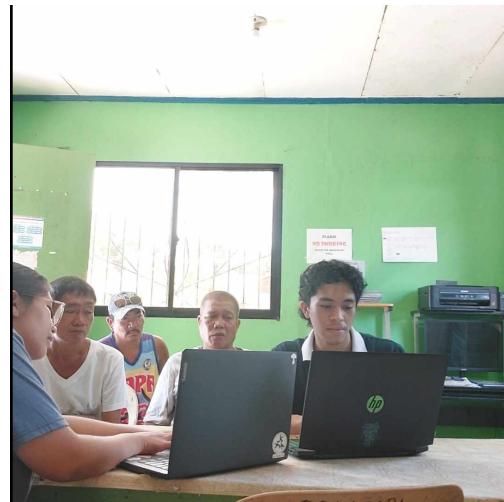


Figure D.1: Respondents (Fishermen, supplier, and retailer) from Barangay Sapa, Miagao



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