



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Research Commons

<https://researchcommons.waikato.ac.nz/>

Research Commons at the University of Waikato

Copyright Statement:

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

The thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of the thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from the thesis.



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Distributed Agent-Centric System for Indigenous Data Sovereignty

Setephano Noovao

Department of Software Engineering
The University of Waikato

Supervisor: Te Taka Keegan

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Engineering in
Software, The University of Waikato, 2024.

Abstract

The sovereignty of indigenous data is a key issue facing indigenous communities. Holochain, a distributed ledger technology, has the potential to address some of the challenges indigenous communities face with respect to data sovereignty. The purpose of this thesis is to critically evaluate the feasibility of using holochain technology to support indigenous data sovereignty. This paper first explores the literature on holochain technology, examining its underlying architecture and features. The research employs a three-phase design science research methodology, conducting an experiment to explore the integration of a decentralized application, holochain, with a prominent cloud service provider. The first phase concentrates on emulating real data-sharing platforms that may utilize data at its disposal, following the principles of indigenous data sovereignty, where a centralized platform could be hosted on a cloud computing service with multiple services in operation. The second phase involves developing a holochain application guided by the principles of indigenous data sovereignty. The final phase seeks to combine the established centralized platform from the first phase with the outcomes of the holochain application from the second phase to experimentally assess whether both technologies can create a data sovereignty solution that aligns with the needs of indigenous data sovereignty. The results of the experiment will provide a solid foundation for understanding the current state of the field and identifying areas where holochain can potentially offer solutions. Furthermore, the results of our research indicate that holochain shows significant promise in tackling the issue of indigenous data sovereignty, although there are considerable limitations that must be addressed and resolved to achieve improved results.

Acknowledgments

My sincere gratitude goes to Professor Te Taka Keegan for his academic guidance as my supervisor and to Professor Steve Reeves for his invaluable insight into my thesis. Their support has been crucial to the completion of this work. I am particularly grateful to my mother, Mareta Opo, for her steadfast love which has enabled me to be here today, and for instilling in me a profound understanding and respect for the significance of indigenous culture. Finally, to my beautiful wife Ellen, saying this wouldn't have been possible without you is no exaggeration—thank you immensely.

Table of Contents

Abstract	i
Acknowledgments	ii
List of Figures	vi
List of Tables	viii
List of Procedures	ix
Acronyms	xi
Glossary	xii
1 Introduction	1
1.1 Rationale and Significance	2
1.1.1 Rationale	2
1.1.2 Significance	2
1.2 Research Methodology	3
1.3 Research Questions	3
1.4 Motivation	4
1.5 Aims and Objectives	4
1.6 Thesis Outline	5
2 Literature Review	6
2.1 Data Sovereignty	6
2.2 Indigenous Data Sovereignty	8
2.2.1 Māori Sovereignty: Understanding and Significance	11
2.2.1.1 The Māori Sovereignty Principles	11

2.2.1.2	Importance of Representing Māori Sovereignty Principles in the Modern Age	13
2.2.1.3	The Impact of Representing Māori Sovereignty Principles on Global Data Governance	13
2.2.1.4	Embracing Māori Sovereignty Principles in Technology and Innovation	14
2.2.1.5	Challenges of Implementation	15
2.2.1.6	Summary	16
2.2.2	Āhau	17
2.2.2.1	Secure Scuttlebutt Protocol	17
2.2.2.2	Implementation	18
2.3	Rust Programming Language	19
2.4	Centralization	20
2.4.1	Cloud Technology and Data Management	22
2.5	Decentralization	22
2.5.1	Peer-to-Peer Network	22
2.5.2	Distributed Hash Table	24
2.5.3	Blockchain	26
2.5.4	Holochain	27
2.5.4.1	Ceptr	28
2.5.4.2	Core Pillars	28
2.5.4.3	Architecture	28
3	Methods and Methodology	30
3.1	Design Science Research	31
3.2	Artifacts	32
3.3	Theory versus Action	32
3.4	Evaluation	32
4	Research Phase 1: Infrastructure Build and Design	34
4.1	Goals and Objectives	34
4.2	Defining Requirements	35
4.2.1	Controlling Access to Data	36
4.2.2	Scalability	36
4.2.3	Interoperability	37
4.2.4	Security and Privacy	37
4.2.5	Data Storage	37
4.3	Prototype Design and Implementation	38
4.3.1	SST	38
4.3.1.1	API Stack	39

4.3.1.2	Authentication Stack	41
4.3.1.3	Database Stack	42
4.3.1.4	SNS Stack	43
4.3.1.5	SQS stack	44
4.3.1.6	Storage Stack	46
4.3.1.7	WebsocketApi Stack	47
4.4	Evaluation	48
5	Research Phase 2 V1: A First Holochain Application	53
5.1	Goals and Objectives	53
5.2	Holochain Formalism	54
5.3	Defining Architecture	58
5.4	Building a Holochain Application	58
5.4.1	IDSOV Iwi Integrity Type Definitions	66
5.4.2	IDSOV Iwi Simple UI Design	68
5.4.3	IDSOV Iwi Build	69
5.5	Challenges and Findings	71
5.6	Evaluation	72
6	Research Phase 2 V2: IDS Holochain Application Prototype	76
6.1	Goals and Objectives	76
6.2	Defining Requirements	78
6.2.1	Understanding Indigenous Data Sovereignty Principles	79
6.2.2	Applying Indigenous Data Sovereignty Principles	79
6.3	Building A Holochain Application - Final Version	81
6.3.1	IDSOV V2 UI Design	84
6.3.2	IDSOV V2 Integrity Types Display	86
6.3.3	IDSOV V2 Functionality	88
6.4	Challenges and Findings	93
6.5	Evaluation	93
7	Research Phase 3: Shared paradigm	95
7.1	Goals and Objectives	95
7.2	Establishing A Connection	96
7.3	Challenges and Findings	99
7.4	Evaluation	101
8	Discussion	103
8.1	Research Questions	103
8.1.1	Can holochain technology be used to support indigenous data sovereignty?	103

8.1.2	What are the key requirements for indigenous data sovereignty and does holochain meet those requirements?	104
8.1.3	What are the potential benefits and challenges experienced when adopting holochain for indigenous data sovereignty?	105
8.2	Limitations	106
8.3	Critical Analysis	108
8.4	Possibilities for Further Research	109
8.5	Conclusion	110
A	Symbols Used In The Thesis	112
A.1	Distributed Systems Symbols	113
A.2	Holochain Distributed Systems Symbols	114
B	Amazon Web Services	116
B.1	AWS API Gateway	116
B.2	AWS Cloudwatch	116
B.3	AWS Cognito	117
B.4	AWS DynamoDB	117
B.5	AWS Lambda	117
B.6	AWS RDS	117
B.7	AWS S3	118

List of Figures

2.1	UDP broadcast packet format [155]	17
2.2	Secure scuttlebutt protocol pub message format	18
2.3	Client to server node placement	21
2.4	Decentralized clients	23
2.5	Data, hash function, key, peer network	25
4.1	Research phase 1 : Central infrastructure directory tree	51
4.2	Research phase 1 : Central infrastructure package	52
5.1	Holochain "hello world" sequence diagram	59
5.2	Research application overview development	60
5.3	Software development process flow	61
5.4	Research phase 2 V1 : Holochain scaffolding tool example	64
5.5	Active "hello world" application with two nodes	65
5.6	Active "hello world" application with multiple entries	65
5.7	Sample "hello world" application Source chain	66
5.8	Sample "hello world" application node graph	67
5.10	IDSOV iwi first design	68
5.11	IDSOV iwi minor improvements	69
5.12	IDSOV iwi application display with entries	70
5.13	IDSOV iwi application display with entries and two agents	71
5.9	Sample "hello world" entries on the entry graph	75
6.1	Adapted changes of the profile	77
6.2	IDSOV V2 Landing Page Frame Design	85
6.3	IDSOV V2 Landing Page Live Application - Dark and Light Mode	86
6.4	IDSOV V2 Successful registration of two agents	87
6.5	IDSOV V2 UI dashboard	88
6.6	IDSOV V2 UI patient record with agent list	91
6.7	IDSOV V2 UI creating a patient record	92

7.1 Holochain Client External HTTP Call in Application Console. 99

List of Tables

2.1	Māori Sovereignty Principles	12
2.2	Comparison of biological and holochain systems naming convention	29
6.1	Adopted alignment of MASov principles with algorithm principles that can translate into requirements for building a holochain application [116]	78
6.2	New Zealand Geographic Region Used In Profile Setup	82
6.3	LinkTypes to artefacts	89
A.1	Distributed Systems Symbols	113
A.2	Holochain Model Attributes [145]	114
A.3	Holochain Model Attributes [145]	115
B.1	AWS Services Overview	116
B.2	AWS Database Types	118

List of Procedures

1	API stack	40
2	Authentication stack	41
3	Database stack	42
4	SNS stack	44
5	SQS stack	46
6	Storage stack	47
7	Websockets stack	49
8	General DNA integrity and coordinator scenario	61
9	Zome functions and entries scenario - create	62
10	Zome functions and entries scenario - create and read	63
13	IDSOV integrity types	67
11	Zome functions and entries scenario - create and update	73
12	Zome functions and entries scenario - create and delete	74
14	V2 Application Integrity Types	83
15	V2 Application Entry and Link Types	84
16	V2 Application Path With Link Usage	89
17	V2 Application DNA Info Implementation	92
18	Rust Websocket Client Listener	96
19	Typescript Websocket Client Listener	97
20	Holochain Client External HTTP Call	98

Acronyms

DHT Distributed hash table. 24, 57

DLT Distributed Ledger Technology. 26

DSR Design Science Research. 31

HC Holochain scaffolding tool. 62

IDS Indigenous Data Sovereignty. 1, 4, 6, 10, 28, 77

P2P Peer-to-peer. 22

SC Source chain. vii, 66

Glossary

Amazon Web Services Amazon Web Services (AWS) is a cloud provider offering fully featured services. These services, which are offered from data centers around the world, are equipped with advanced provisions for businesses and developers. 48, 101, 118, 119

Hapū An indigenous Māori term that refers to several families (whānau) who share the same ancestry in Māori culture. 17, 18

Holochain Application A Holochain application, commonly known as a "hApp," is a decentralized application that utilizes the Holochain framework, which is implemented in the Rust programming language. Holochain presents a distinct approach to distributed computing and decentralized applications, distinguishing it from conventional blockchain technology. 28, 79

Iwi An indigenous Māori term describing the broad collective, the whole tribe or nation, which includes all Hapū and Whānau. 17

Pātaka The term Pātaka is indigenous to the Māori language, and within the context of Āhau, it denotes the database stored on a user's device. In particular, a desktop operating system capable of housing a server. 17

Rust Rust, created by Mozilla research employee Graydon Hoare, is a systems programming language renowned for its emphasis on safety, concurrency, and performance. It was specifically developed to overcome the limitations of other programming languages, especially in terms of memory safety and the prevention of data races in concurrent programs. 19

SST The SST framework referencing the serverless stack, being an open source solution, provides a convenient means of constructing contemporary full-stack applications and infrastructure on the Amazon Web Services (AWS) platform. 36

WASM A byte code format for low-level programs that possesses the capability to run on virtually any platform, including web browsers, is utilized by Holochain. To execute the zones, Holochain

anticipates that they will be compiled into WebAssembly, allowing the ribosome to execute them. 63

Whānau An indigenous Māori expression referring to family, the fundamental social unit in Māori culture. 17

Zome The zome is an essential component of modularity in a Holochain DNA. It consists of a set of zome functions and can be either an integrity or a coordinator zome. 28, 29

Zome Function A function, created by the developer of a zome, that enables external code to access and utilize the features and capabilities of the zome. 29

Chapter 1

Introduction

In today's data-driven society, the significance of data sovereignty, specifically for indigenous communities, has been increasingly acknowledged in recent times [28], [84], [106], [120]. Data sovereignty, particularly Indigenous Data Sovereignty pertains to the entitlement of individuals or communities to possess, govern, and retrieve their own data [84], [106], [116], [133]. The notion is of significant importance for indigenous populations that have endured extensive colonization and marginalization over time [85], [94], [130]. Data sovereignty plays a critical role in empowering indigenous communities by allowing them to regain authority over their own data and safeguard their cultural knowledge, traditions and identity [86], [132]–[134]. Utilizing blockchain technology, specifically holochain, is an intriguing aspect to consider when examining data sovereignty [74]. The decentralized and unalterable characteristics of blockchain technology have the capacity to empower indigenous communities and aid in the establishment of their data sovereignty [105]. Holochain, as a distinctive framework, offers a decentralized method for managing data [80], [91], [100]. The emphasis is changed from focusing on data to focusing on agents, allowing individuals or communities to have complete control over their data without relying on a central authority [61], [70], [113]. This approach is in line with the principles of indigenous data sovereignty, as it gives communities the power to govern and manage their own data according to their own terms [70].

While the theoretical connection between holochain and indigenous data sovereignty is present, there is still limited practical and empirical knowledge about this intersection [74], [100], [105]. A comprehensive literature review within the domain of Software Engineering is essential to map the current landscape, identify gaps, and critically assess the viability and challenges of implementing holochain for IDS. This review will not only synthesize existing knowledge, but also highlight the technological, ethical, and governance considerations specific to this integration. The nuanced understanding gained from this review is expected to provide a foundational basis for future empirical research, technological development, and policy formulation regarding the use of holochain for indigenous data sovereignty.

1.1 Rationale and Significance

Indigenous peoples have been marginalized and taken advantage of in the area of data collection and use for a long time [84], [106], [116], [133]. Indigenous research sovereignty is connected to and builds on the movement for indigenous data sovereignty, which encourages indigenous control of their data, including their collection, ownership, and application [57], [106], [133]. This movement acknowledges the natural rights and interests of indigenous peoples in their data and works to protect their cultural heritage, knowledge, and autonomy [33], [84], [106]. Holochain, a promising distributed ledger technology, could address some of the issues indigenous communities face with respect to data sovereignty. Holochain is a scalable and agent-centric distributed ledger technology that allows decentralized data storage and peer-to-peer communication [105].

1.1.1 Rationale

Holochain is a paradigm shift in the field of decentralized technologies [105]. Holochain, in contrast to traditional blockchain systems, adopts an agent-centric approach that provides individuals or communities with the ability to have control and independence over their data and interactions [74], [100]. The decentralized structure of this system perfectly aligns with the fundamental principles of indigenous data sovereignty, allowing indigenous communities to exercise control over their data autonomously, without relying on centralized entities [105].

In our modern era, the protection of data security and privacy is of utmost importance [82], [124], [125]. Indigenous communities often have valuable cultural and traditional knowledge that requires protection against unauthorized access or misuse [116], [133]. The implementation of cryptographic methods and local data storage by holochain improves security and privacy measures, reducing the likelihood of data breaches and unauthorized access [100]. The size and complexity of data from indigenous communities can differ greatly [33]. It is crucial to have efficient scalability in order to accommodate this diversity [50], [84]. The architectural design of holochain enables effortless scalability by avoiding the need for global consensus mechanisms [100], [124]. Instead, each agent independently handles its own data, reducing computational burden and ensuring system responsiveness even when dealing with larger datasets [89]. Indigenous data sovereignty goes beyond data management; it includes the empowerment of indigenous communities to make well-informed choices about their data and how they are used [74]. Holochain's peer-to-peer system promotes self-governance within communities, enabling groups to create rules for sharing data and oversee interactions among community members [125].

1.1.2 Significance

Preservation of Cultural Heritage: Knowledge, traditions, and languages of indigenous cultures are extremely valuable, but often in danger of being lost or diluted. The preservation of this cultural heritage is closely related to the concept of data sovereignty [105]. The use of holochain has the

potential to empower indigenous communities in the management of their data, which is crucial to preserving and passing on their cultural legacy to future generations [74].

Advancing Technology for Social Equity: Enhancing the sovereignty of indigenous data through the integration of innovative technologies is an essential measure to narrow the digital gap [31], [38], [50]. This study adds to the ongoing conversation about the use of state-of-the-art technology to empower marginalized and underrepresented groups, in line with the larger goals of technological progress and social equity.

Academic Contribution: This research addresses a crucial gap in the contemporary academic literature by responding to the call to action and conducting a review of the literature on indigenous data sovereignty and a decentralized technology such as holochain [105]. This study specifically examines the overlap and intersection between māori sovereignty and holochain, the core principles of an agent-centric technology, its technical advantages, costs, and potential. Additionally, this thesis aims to explore the availability and accessibility of data within the holochain framework compared to the centralized paradigm. By analyzing the literature, this study lays the groundwork for future research and dialogue on the relationship between indigenous data sovereignty and an agent-centric technology like holochain.

In summary, the justification for investigating holochain in the context of indigenous data is firmly based on its ability to redefine data handling, improve security and privacy, empower communities, and protect indigenous cultural heritage. The importance of this research goes beyond technological progress and also includes social justice and the understanding of these crucial issues by the academic community. In the following sections, we will further examine the methodology, literature review, and findings, expanding on this groundwork to thoroughly investigate the subject of indigenous data sovereignty in the context of holochain and software engineering.

1.2 Research Methodology

To investigate the potential of holochain in ensuring the autonomy of indigenous data, a design science research approach will be adopted. This methodology focuses on creating novel solutions to real-world problems through successive design and assessment cycles [63], [66]. It involves the development and testing of artifacts, such as software systems or frameworks, to meet particular research objectives [34], [88]. The initial step in the design science research process will be to perform an extensive review of the literature on the sovereignty of indigenous data and the difficulties faced by indigenous communities in data collection and use. This review of the literature will provide a strong basis for understanding the current state of the field and recognizing areas where holochain can potentially provide solutions.

1.3 Research Questions

The research questions addressed in this thesis are as follows:

1. Can holochain technology be used to support indigenous data sovereignty (Indigenous Data Sovereignty (IDS))?
2. What are the key requirements for indigenous data sovereignty and does holochain meet those requirements?
3. What are the potential benefits and challenges experienced when adopting holochain for indigenous data sovereignty?

1.4 Motivation

This thesis seeks to investigate the potential of holochain technology to empower indigenous communities in terms of data sovereignty. It is driven by the conviction that indigenous people should have the right to decide how their data are collected, stored, and shared, and that holochain technology may provide a means to achieve this. The aim is to explore the challenges facing indigenous communities in terms of data sovereignty and to assess the potential of holochain technology to address these issues.

Although holochain may offer potential solutions to address the challenges of data sovereignty for indigenous communities, it is important to consider the potential challenges and limitations that may arise. One of the main concerns is accessibility and inclusion of holochain technology for indigenous communities. Although holochain offers decentralized and distributed data management, it requires technical knowledge and infrastructure for implementation. Many indigenous communities may lack the resources and expertise necessary to adopt and use holochain effectively. This could create a digital divide and further marginalize these communities in the realm of data governance.

1.5 Aims and Objectives

This thesis aims to evaluate whether holochain technology can be used to support indigenous data sovereignty. The objectives of this thesis are the following.

1. Review and analyze the literature on IDS and holochain technology.
2. Identify the key requirements for IDS and assess whether holochain meets those requirements.
3. Conduct an experiment to evaluate the practicality and effectiveness of using holochain technology for indigenous data autonomy. The experiment will involve developing a holochain application, incorporating ongoing enhancements as we gain insight from each phase, with the goal of facilitating data exchange between holochain and a centralized system.

We will aim to answer the research questions through a comprehensive review of the literature on indigenous data sovereignty, exploring the concepts, principles, and challenges associated with it.

1.6 Thesis Outline

- Introduction: This chapter will give an overview of the research topic, including its history, inquiries, inspiration, goals, and objectives.
- Review of the literature: This chapter will review and analyze the relevant literature on indigenous data sovereignty and holochain technology.
- Methods and Methodology: This chapter will describe the research methods and methodology used in the study, including data collection and analysis techniques.
- Research Phases 1-3: These chapters will build on the prototype of holochain and the centralized infrastructure, followed by the outcome.
- Discussion: This chapter will present the findings based on the research phases conducted, highlighting the key concepts and challenges related to indigenous data sovereignty, as well as the features and capabilities of holochain technology.

Chapter 2

Literature Review

In this literature review, we examine the potential of holochain in safeguarding the independence of indigenous data. Through an exploration of current research and real-life examples, our objective is to provide a thorough understanding of how holochain can be utilized to assist indigenous communities in asserting their rights over data and safeguarding their cultural legacy. Furthermore, we will assess the technical functionalities of holochain and its alignment with the principles and requirements of indigenous data governance.

In this review, our aim is to contribute to the ongoing discussion of indigenous data sovereignty and to offer perspectives on how holochain can be a potential solution in the field of software engineering. The empowerment of indigenous communities and the preservation of their cultural knowledge and identity are heavily based on data sovereignty. Furthermore, with the continuous influence of technological progress, it is crucial to explore creative approaches that uphold the principles of indigenous data sovereignty.

This thesis investigates two topics: holochain technology [59] and IDS. The review of holochain technology looks at the fundamentals and structure of the technology, its key characteristics, and its potential uses in different fields. In addition, it examines the benefits and drawbacks of using holochain for data sovereignty. The indigenous data sovereignty review examines the principles and objectives of the movement, the difficulties faced by indigenous communities in asserting their rights to data sovereignty, and the existing solutions and frameworks proposed to address these issues.

2.1 Data Sovereignty

Data sovereignty is the notion of having control over one's own data [105], [106]. It is the right to decide how data is collected, stored, processed, and shared. This concept has become increasingly relevant in recent times due to the rise of digital technologies and the extensive collection and use of personal data [30], [38], [50], [105], [106]. Scholars have acknowledged the importance of data sovereignty as a component of state sovereignty in the digital age. Data sovereignty can be examined

from a variety of angles, including technical, legal, and cultural [84], [106], [133].

From a technical point of view, data sovereignty is concerned with the technology and processes that guarantee data control [133]. This includes considerations such as where data is stored, encryption of data, and access restrictions [106], [133]. From a legal point of view, data sovereignty involves the implementation of laws and regulations that protect individuals' rights to their data and regulate how it is collected and used [33], [84]. From a cultural point of view, data sovereignty recognizes the importance of indigenous communities and their right to manage their own data [49], [57], [106]. The sovereignty of indigenous data is especially notable, as it highlights the special rights and interests of indigenous people in relation to the collection, ownership, and utilization of their data [106], [116], [120].

The concept of data sovereignty has received significant attention in the literature, and scholars have explored various dimensions and implications of this concept [33], [84], [85]. In the context of indigenous data sovereignty, the authors have emphasized the need for decolonization and recognition of indigenous rights to data [106]. They argue that indigenous communities should have the power to determine who is counted among them and what data are collected, reflecting their interests, values, and priorities [57], [104]. Furthermore, scholars have highlighted the importance of consent and ensuring benefits for those who are data subjects [86]. They have also emphasized the need to address the issues of data storage, access, and ownership in a way that respects indigenous customs, practices, and laws [83], [133].

The need to seriously consider data sovereignty arises from the historical exploitation and marginalization of indigenous communities [50]. Indigenous communities have long been subjected to the extraction and misuse of their data by external entities, perpetuating power imbalances and violating their rights to self-determination and self-government [106], [134]. Data sovereignty provides a means for indigenous communities to regain control over their data, protect their privacy, and determine how their data are used and shared [106]. By prioritizing data sovereignty, we can assist the historical injustices and power imbalances that have affected indigenous communities [105]. Furthermore, data sovereignty is relevant not only to indigenous communities, but also to any individual or organization that values privacy and security [85], [116]. The concept of data sovereignty is important in the context of privacy and security, as it ensures that individuals and communities have control over their own data. This control is crucial to preventing unauthorized access, data breaches, and exploitation [27]. In summary, data sovereignty is a concept that emphasizes the right of individuals and communities to exercise ownership and control over their data. By acknowledging and respecting data sovereignty, we can promote equity, justice, and self-determination for indigenous communities.

Certain key concepts within data sovereignty include:

- **Ownership:** Data sovereignty emphasizes that individuals and organizations have control over their data. They have the authority to decide how their data are gathered, kept, handled, and distributed [69].
- **Jurisdiction:** Data sovereignty suggests that data should be managed in accordance with the laws

and regulations of the country in which it is stored or processed [69].

- **Security:** Data sovereignty stresses the requirement for robust data protection protocols to prevent unauthorized access or violations [69].
- **Privacy:** The importance of protecting individual privacy rights and adhering to the relevant privacy regulations when dealing with data is emphasized by data sovereignty [110].
- **Consent:** Data sovereignty emphasizes the importance of individuals giving their consent to have their data collected, kept, and used [69], [110].
- **Portability:** Advocates of data sovereignty call for the ability to conveniently shift and move data between different suppliers or legal systems [98], [110].
- **Accountability:** Individuals and organizations are responsible for the proper use and safeguarding of their data under the concept of data sovereignty [98].

The significance of data sovereignty is clear. It is a way to address past wrongs and empower indigenous communities by giving them control over their data [95]. This allows them to tell their own stories and shape their narratives, ensuring that their perspectives and experiences are accurately represented [23]. Additionally, data sovereignty is essential to protect the privacy and security of individual data. It recognizes the rights of indigenous peoples to exercise ownership and control over their data, including data collection, ownership, and application [95]. This enables them to make decisions about how their data are collected, used, and shared, allowing them to determine the priorities and goals of data collection and ensure that the data are used in a way that benefits their communities and supports their development goals [95].

Data sovereignty has a significant impact on the way data is viewed and utilized. It challenges the traditional notion of data as a commodity that can be exploited for financial gain, rather than highlighting its value as an asset that belongs to individuals and communities and should be used in accordance with their values and interests [69], [95], [110]. In addition, it gives people more control over how their data are collected, used, and shared, allowing them to make informed decisions about data collection and use [110]. Additionally, data sovereignty is shifting the perception of data from a purely technical or economic resource to a social and cultural one, recognizing that data is not a neutral entity, but rather embedded within social contexts and power dynamics [106], [110], [133]. Finally, it encourages the development of ethical frameworks and guidelines for data governance, ensuring that data practices are in line with the principles of privacy, consent, and fairness [133].

2.2 Indigenous Data Sovereignty

Indigenous data sovereignty has become a widely discussed topic in recent times. It is based on the right of indigenous peoples and nations to have control over the collection, governance, ownership,

and use of data related to their people, lands, and resources [30], [38], [106]. This concept is rooted in indigenous understandings of sovereignty, which contrast with the dominant views on "data sovereignty" and current practices. Indigenous data sovereignty emphasizes the need to recognize and honor the inherent rights and interests of indigenous communities in relation to the data collected about them and their territories [84].

Although the concept of indigenous data sovereignty is rooted in the rights and interests of indigenous communities, it also has broader implications for data governance and decision making, in general [95]. Indigenous data sovereignty is not just about data ownership, but also includes the right of indigenous peoples to determine how data are collected, accessed, analyzed, interpreted, managed, disseminated, and reused [95], [106]. This includes the right to give or withdraw consent for data collection, the right to control access to data, and the right to use data for their own self-determination and development. Indigenous data sovereignty is intertwined with indigenous peoples' inherent rights of self-determination and governance over their communities, territories, and resources. Indigenous data sovereignty is important for several reasons [84], [95], [106].

Initially, indigenous data sovereignty acknowledges the historical and ongoing exploitation of data belonging to indigenous communities by external agencies and institutions. In addition, it acknowledges the unique knowledge systems and perspectives that indigenous communities contribute to data collection and analysis [86], [106]. In addition, it seeks to empower indigenous communities by promoting their autonomy and agency in decision-making processes that impact their lives and territories [95]. In addition, its goal is to challenge the power dynamics and colonial legacy that have historically marginalized and excluded indigenous peoples [134]. Ultimately, indigenous data sovereignty offers a framework for recognizing and respecting the rights and interests of indigenous communities in relation to data, which is crucial for promoting justice, equity, and self-determination in the collection, management, and use of data [104]. In the context of data sovereignty, indigenous communities face unique challenges due to the historical exploitation of their culture and knowledge. Some of these challenges include:

- **Historical Exploitation:** Indigenous communities have a long history of exploitation and cultural appropriation, particularly in the context of data and knowledge [57].
- **Lack of representation:** Indigenous communities often face a lack of representation and agency in making decisions about their data [86], [133].
- **Limited access to resources:** Indigenous communities may have limited access to the resources and infrastructure necessary to effectively govern their data [85].
- **Cultural sensitivity:** Indigenous data contain cultural knowledge and traditional practices that require special considerations for protection and preservation [84].
- **Inadequate legal frameworks:** Existing legal frameworks may not adequately address the unique challenges indigenous communities face with respect to data sovereignty [95], [105], [120].

Several cases highlight the importance of data sovereignty for indigenous communities. In an instance involving indigenous Australian communities, researchers collected blood samples for genetic research without obtaining the appropriate consent or informing the participants. This led to a breach of trust and infringed upon the sovereignty of indigenous data [50], [132]. Another study within the context of Aotearoa New Zealand is where the māori community has been advocating for greater sovereignty over their data and the recognition of indigenous rights to control and manage their data [84]. To address these injustices, indigenous communities have been advocating for the development of research protocols and principles that honor and protect the sovereignty of indigenous data [95].

In New Zealand, the māori Data Sovereignty Network in reference to māori has been a leader in advocating for indigenous data sovereignty [84]. This framework provides a way to advance data governance practices that prioritize the rights and autonomy of indigenous peoples [133]. These principles, rooted in the concepts of ownership, control, access and possession (OCAP), aim to address historical imbalances and tensions between protecting indigenous rights and supporting open data initiatives [77]. They have emphasized the need for effective definitions and the development of protocols and principles that respect indigenous rights. Indigenous people must have authority over their data to ensure their ethical and respectful use, as well as the preservation of cultural knowledge and intellectual property [30], [77], [95].

By asserting indigenous data sovereignty, communities can regain control over their data and ensure that they are collected, used, and shared in ways that align with their cultural values, aspirations, and self-determination [95]. Controlling and acquiring data is a fundamental objective for indigenous communities, as it plays a critical role in achieving self-determination and building an information-resilient society [31]. The Declaration of the United Nations on the rights of indigenous peoples acknowledges the importance of maintaining, managing, protecting, and advancing information and data [23]. This article [106] seeks to provide information on the national dialogue on indigenous data sovereignty in Aotearoa, New Zealand, while raising awareness of implementation strategies and difficulties. Examining existing laws that regulate statistical data offers a special opportunity to gain valuable insight from various points of view concerning the realization of indigenous data sovereignty principles [24].

IDS represents an effective approach to establish consent, respecting rights, and ensuring benefits for people who are the subjects of collected data. More specifically, the concept addresses inequities and exploitation concerning those who have had limited control over their own personal or collective information around multiple parts of the world via numerous working groups consisting of capable scholars with this specialization [77], [133]. Furthermore, indigenous data sovereignty encompasses the right of indigenous communities to determine how their data are collected, accessed, analyzed, interpreted, managed, disseminated, and stored [106]. This includes the right to establish their data governance structures and practices, as well as the ability to negotiate partnerships and agreements with external entities for data sharing and collaboration [84].

2.2.1 Māori Sovereignty: Understanding and Significance

The objective of controlling and obtaining data is crucial for numerous indigenous communities, since it is a fundamental aspect of self-governance and, consequently, necessary to establish a society that is resilient to information. The principles of māori sovereignty, which are based on the concepts of rangatiratanga (authority) and kaitiakitanga (guardianship), play a significant role in advocating for the rights and interests of indigenous communities, such as the māori people [84], [116]. These principles acknowledge the inherent power and duty that māori hold in relation to their own data and information. Māori sovereignty principles acknowledge the significance of data as a taonga (treasure) and an essential element of mātauranga māori. These principles affirm the rights of hapu (sub-tribes) and iwi (tribes), both collectively and individually, and emphasize the importance of managing and utilizing data in a manner that benefits the māori community as a whole [30], [106]. In the modern era of technology, the significance of data has grown significantly in different domains such as policymaking, research, and business growth. Consequently, it is essential to adhere to the principles of māori sovereignty to guarantee that the gathering, administration, and utilization of data are in accordance with the values and goals of the māori community [84], [95]. By integrating māori sovereignty principles, such as rangatiratanga and kaitiakitanga, into the processes of decision-making and governance structures, it is possible to establish an atmosphere that honors māori values and acknowledges their control over their own data [84]. This approach not only fosters inclusivity and fairness, but also guarantees that the experiences and viewpoints of indigenous communities are adequately acknowledged and considered. To sum up, the principles of māori sovereignty have a crucial function in representing and safeguarding the rights and interests of indigenous communities, specifically the māori, regarding data sovereignty. By integrating these principles into laws and policies, we can strive to achieve data sovereignty for indigenous communities worldwide.

2.2.1.1 The Māori Sovereignty Principles

The māori sovereignty principles hold significance in contemporary society as they safeguard the rights and self-governance of indigenous communities, particularly the māori community, in relation to data sovereignty. These principles are also deeply connected to other indigenous sovereignty movements globally [84]. The statement acknowledges the māori control over their own data and information, ensuring that it is handled and utilized in a manner that is consistent with māori principles, goals, and the communal and personal rights of hapu and iwi [84]. In the modern era, as technology becomes more prevalent, it is essential to emphasize the importance of māori sovereignty principles [28]. These principles play a vital role in allowing indigenous communities to have authority over their data and actively engage in decision-making processes that impact them [106]. The key māori sovereignty principles are outlined below:

1. **Rangatiratanga:**

This principle emphasizes self-determination, autonomy, and authority. It recognizes the rights

Māori Sovereignty Principle	Translation
Rangatiratanga	Authority
Whakapapa	Relationships
Whanaungatanga	Obligations
Kotahitanga	Collective Benefits
Manaakitanga	Reciprocity
Kaitiakitanga	Guardianship

Table 2.1: Māori Sovereignty Principles [106]

of māori communities to have control over their own data and information and the ability to make decisions that align with their cultural values and aspirations [116].

2. **Whakapapa:**

Whakapapa pertains to the interconnection and genealogical associations present within māori societies. It underscores the significance of acknowledging and honoring the ties and relationships among individuals, families, communities, and the land. It acknowledges that data encompasses not only a compilation of facts but also embodies the narratives, past events, and cultural value of the māori population [116].

3. **Whanaungatanga:**

The importance of relationships and kinship connections in māori communities is emphasized by this principle. It highlights the need to work together and form partnerships with different stakeholders, including government agencies, researchers, and data custodians, to make decisions that respect and uphold māori values and goals [116].

4. **Kotahitanga:**

This principle encompasses the idea of unity and collaboration. It highlights the importance of collective effort and solidarity within māori communities, as well as the need to establish partnerships and engage in joint work with non-indigenous organizations [116].

5. **Manaakitanga:**

This principle includes the concepts of welcoming, attending to needs, and fostering growth. It highlights the significance of utilizing data in a way that supports favorable results and the welfare of māori communities. It also guarantees that data is gathered, handled, and utilized in manners that show respect, adhere to ethical standards, and bring benefits to all parties involved [116].

6. **Kaitiakitanga:**

This principle pertains to the concept of guardianship and stewardship. It highlights the duty of māori communities to safeguard and conserve their data, as well as the natural environment and

resources that are intertwined with it. When examining and delving into the principles of māori sovereignty, it becomes clear that these principles embody the fundamental values, ambitions and entitlements of indigenous populations. By advocating for self-governance, independence, and power, these principles acknowledge and validate the sovereignty of māori communities, acknowledging their inherent right to govern their own data and make choices that align with their cultural values and aspirations [116].

The principles of māori sovereignty encompass the fundamental values, aspirations, and rights of indigenous populations. These principles emphasize the importance of self-government, independence and power, acknowledging and validating the sovereignty of māori communities. They recognize the inherent entitlement of these communities to control their own information and make choices that align with their cultural values and aspirations [106], [116].

2.2.1.2 Importance of Representing Māori Sovereignty Principles in the Modern Age

In today's era, it is of utmost importance to embody the principles of māori sovereignty for various reasons. Firstly, it is crucial to acknowledge and honor the rights and aspirations of indigenous communities [57], [86]. By incorporating and recognizing māori sovereignty principles, we can guarantee that the viewpoints and outlooks of māori societies are acknowledged and appreciated during the decision-making process [84]. This promotes fairness and equality, as it empowers indigenous peoples to have a voice in matters that directly affect their lives and welfare [94]. Moreover, representing māori sovereignty principles is significant in safeguarding and conserving cultural heritage. By implementing these principles, we can guarantee the protection and rejuvenation of the māori language, customs, and practices. This aligns with the broader objective of decolonization, as it challenges the dominance of Western systems and promotes the recognition and validation of indigenous knowledge and ways of understanding [49], [84]. Moreover, the representation of māori sovereignty principles in contemporary times is crucial for promoting reconciliation and healing. It signifies a step towards addressing the historical injustices and marginalization that indigenous communities have endured [95]. By acknowledging and respecting māori sovereignty principles, we acknowledge and confront the enduring impact of colonization and strive towards constructing more inclusive and fair societies [33]. Additionally, the representation of māori sovereignty principles in the modern era is essential for advancing sustainable development and environmental stewardship [84].

2.2.1.3 The Impact of Representing Māori Sovereignty Principles on Global Data Governance

The inclusion of māori sovereignty principles in contemporary times greatly influences global data governance [133]. Recognizing and respecting the sovereignty of māori communities and their inherent authority over their data establishes a precedent for embracing different cultural viewpoints in data management and utilization [84]. This not only fosters fairness and equality at the community level, but also contributes to a more comprehensive and considerate global data environment [95].

By integrating māori sovereignty principles into international conversations about data governance, the current dominance of Western-centric systems is challenged, and the recognition of various indigenous knowledge and ways of understanding is promoted [77], [95]. This shift towards a more culturally sensitive approach to data governance emphasizes the importance of valuing and incorporating different cultural values and aspirations. The implications of this are substantial in the global digital landscape, as it requires existing data governance frameworks to be adapted in order to accommodate diverse cultural perspectives and promote the inclusion of indigenous communities [77], [133].

Nevertheless, similar to any substantial change in perspective, the inclusion of māori sovereignty principles in worldwide data governance encounters difficulties. It necessitates thoughtful conversations and evaluations concerning the uniformity and compatibility of data among various cultural and ethnic communities, as well as potential clashes with current international data governance structures [38], [130]. Furthermore, the incorporation of indigenous data governance models into a globalized digital environment requires a sensitive equilibrium between acknowledging indigenous rights and ambitions and guaranteeing the overall effectiveness and interconnectedness of data systems on a global level [95].

In addition to the difficulties in managing and using data, there are complex legal and ethical consequences that need to be carefully considered [84]. The combination of indigenous data sovereignty and international data governance frameworks gives rise to concerns about legal uncertainties, conflicts, and the practical viability of establishing separate governance models for indigenous data within a globally interconnected infrastructure [95]. The inclusion of māori sovereignty principles in the discussion on data governance provides valuable insights and perspectives. However, it is important to approach this topic with sensitivity, taking into account the complex interplay of cultural, legal, and ethical factors in the ever-changing digital world [86]. Balancing indigenous data sovereignty with global interconnectedness is a crucial challenge that requires collaborative discussions and innovative solutions [57].

Therefore, the incorporation of māori sovereignty principles in contemporary times not only influences data governance at the local level, but also offers a chance to reform global data governance frameworks in a way that recognizes and honors various cultural values and aspirations. This represents a significant shift toward promoting a more comprehensive, fair, and culturally aware approach to managing and utilizing data worldwide.

2.2.1.4 Embracing Māori Sovereignty Principles in Technology and Innovation

In addition to its influence on worldwide data governance, the adoption of māori sovereignty principles also applies to the field of technology and innovation [30]. By incorporating māori viewpoints and values into technological advancements, there is a chance to foster inclusiveness, safeguard cultural heritage, and address ethical concerns in the creation and utilization of novel technologies [84]. By integrating māori sovereignty principles into technology and innovation, a more culturally aware approach to product development and digital solutions is promoted. This approach emphasizes the

inclusion of indigenous viewpoints in data-driven advancements, guaranteeing that technology aligns with and honors varied cultural values and goals [50].

Moreover, incorporating māori sovereignty principles into technology and innovation can result in the development of ethically guided technological solutions. By prioritizing autonomy, self-determination, and cultural empowerment, advancements in technology can be designed and executed with the goal of benefiting and empowering indigenous communities [106], [116]. This approach not only acknowledges the sovereignty of māori communities but also contributes to the ethical progress and implementation of technology worldwide. Simultaneously, the inclusion of māori sovereignty principles in technology and innovation poses difficulties in terms of integrating various viewpoints and values into technological solutions [104]. This integration may also result in conflicts with established technological paradigms and raise ethical concerns regarding the incorporation of indigenous knowledge into digital advancements. Achieving a balance between a forward-thinking approach and cultural sensitivity, as well as ethical responsibility, becomes crucial when harnessing the wisdom of māori sovereignty principles in technology and innovation [31].

In general, the adoption of māori sovereignty principles in technology and innovation represents a significant change towards fostering an inclusive technological environment that values diverse cultural beliefs and acknowledges the independence of indigenous communities. By integrating technological progress with the principles of māori sovereignty, it creates an opportunity for a more ethically aware, culturally considerate, and inclusive technological sphere.

2.2.1.5 Challenges of Implementation

Recognizing the significance of māori sovereignty principles, it is crucial to note that there are differing perspectives on the integration of these principles into legislation and policy frameworks. Certain individuals and groups contend that emphasizing the sovereignty of indigenous communities, particularly in relation to data, could present difficulties in terms of ensuring consistency and compatibility of data across various cultural and ethnic groups [38], [85]. They posit that having distinct data governance frameworks for different communities may result in fragmentation and inefficiencies in data administration and utilization, impeding the potential for widespread innovation and advancement. Moreover, there is apprehension that prioritizing māori sovereignty principles in contemporary times might sustain separatism and impede endeavors towards societal cohesion and integration [56]. Detractors contend that advocating for distinct governance and decision-making procedures for particular ethnic or indigenous communities could result in exclusion and a lack of inclusiveness within larger societal frameworks [84]. They underscore the possibility of conflict and disunity if various groups seek separate authority and administration of data within a common digital environment [57].

Furthermore, there are critics who have concerns regarding the practical implications of maintaining māori sovereignty principles in a digital landscape that is globalized. These critics question the feasibility of implementing unique governance models for indigenous data in a globally connected and interdependent data infrastructure [77], [104]. They also raise concerns about the potential complexities, conflicts, and legal uncertainties that may arise when indigenous data sovereignty

intersects with international data governance frameworks [106]. These concerns are emphasized by individuals who hold different viewpoints. The importance of māori sovereignty principles in highlighting matters of justice, self-determination, and the preservation of cultural heritage cannot be overstated. However, it is crucial to have detailed discussions that take into account the potential difficulties and consequences that may arise from prioritizing indigenous data sovereignty in legislative and policy settings. Finding a balance between acknowledging indigenous rights and aspirations and considering broader concerns such as standardization, inclusivity, and global interconnectedness can pose intricate challenges that require careful analysis and thoughtful consideration [84], [116].

2.2.1.6 Summary

In summary, the portrayal of māori sovereignty principles in contemporary times offers both advantages and difficulties that necessitate thorough examination and discussion. It is crucial to acknowledge and honor the rights and ambitions of indigenous communities, while also addressing concerns about potential divisions and inefficiencies in data management and usage. Achieving a balance between preserving cultural heritage and fostering inclusivity within broader societal frameworks is a challenging endeavor that requires careful deliberation and subtle strategies.

In order to progress, it is crucial to participate in collaborative and inclusive procedures that include indigenous communities, policymakers, and stakeholders from various backgrounds. This entails discovering shared interests and creating solutions that respect the authority of indigenous data, while also considering the practical consequences of implementation in a globalized digital environment. By promoting meaningful conversation and exploring inventive approaches to reconcile divergent viewpoints, it is feasible to navigate the intricacies associated with representing māori sovereignty principles in the contemporary era.

By promoting meaningful communication and exploring creative approaches to reconcile divergent viewpoints, it is feasible to navigate the complexities and intricacies linked to the representation of māori sovereignty principles in the contemporary era. In the end, acknowledging and integrating māori sovereignty principles are crucial to advance social equity, protect cultural legacy, and promote reconciliation. Furthermore, by adopting these principles, societies can strive to construct more equitable and inclusive structures that appreciate diversity and allow indigenous communities to control their data according to their cultural values and aspirations. This undertaking not only upholds the autonomy of māori communities but also promotes the ethical advancement and implementation of technology worldwide.

In order to advance social justice, preserve cultural heritage and promote reconciliation, it is crucial to acknowledge and integrate māori sovereignty principles. By embracing these principles, societies can strive to create fair and inclusive systems that appreciate diversity and empower indigenous communities to govern their data according to their cultural values and aspirations.

2.2.2 Āhau

Āhau ¹ is a decentralized application focused on indigenous data sovereignty [137], with data stored locally on users' devices and optionally using a Pātaka. The Pātaka functions as a database on the user's preferred desktop operating system. It enables data backup and synchronization between the user and their Hapū, Whānau (family, subtribe), and the broader Iwi (tribe) [137]. Āhau's goal is to ensure data sovereignty for Whānau and communities, maintaining their cultural values and significance. In addition, the Āhau application utilizes cryptographic methods including key exchange protocols and digital signatures to enhance the security of user communications, effectively preventing unauthorized entities from intercepting or altering the transmitted data.

2.2.2.1 Secure Scuttlebutt Protocol

Āhau utilizes the secure scuttlebutt protocol ², a peer-to-peer event sharing system originally intended for social applications [54]. Each user is recognized by a cryptographic keypair ed25519 generated at the start of the application. This process creates the keypair and stores it in a directory on the user's device. Once the identity is established, a user can then interact with other peers, but they need the public key and address of another peer. Peer discovery often involves knowing the TCP/IP address and port as seen in 2.1, but as peers broadcast UDP packets every second on their local network, this allows local network peers to communicate without needing internet access [155].

net:192.168.1.123:8008~shs:FCX/tsDLpubCPKKfIrw4gc+SQkHcaD17s7GI6i/ziWY=

Figure 2.1: UDP broadcast packet format [155]

Furthermore, another method of discovery amongst peers are invite codes. The invite codes contain the pub's domain name, port and public key. The pub refers to a user's advertising message which are useful in allowing peers to find you and are more or less for users' who know other users' on the existing platform. The messages generally have the following format in 2.2. The pub format stipulates a user *Alice* indicating that they recognize the pub belonging to *Bob* with the pub's domain name and port, in the context of our sample in 2.2, the user *Alice* is the *author* and *Bob* belongs to the *key* under the *address* property. The pub utilizes a distinctive gossip-based algorithm to disseminate messages across the network, guaranteeing efficient and reliable transmission while preserving data integrity [103]. This method improves scalability and robustness in message distribution within the

¹<https://ahau.io/>

²<https://scuttlebutt.nz/>

Secure Scuttlebutt ecosystem, enhancing its decentralized architecture's ability to withstand single points of failure. By leveraging this novel broadcast mechanism, the protocol cultivates a vibrant and interconnected social network where users can safely engage and exchange information without risking privacy or security. The integration of this messaging system with the protocol's cryptographic safeguards secure peer-to-peer communication networks, providing users a reliable platform to interact with others while maintaining control over their data and communications.

```

1      {
2          "author" : "@XYZ/tsDLpubCPKKfIrw4gc+SQkHcaD17s7GI6i/ziWY=.ed25519",
3          "content" : {
4              "type" : "pub",
5              "address" : {
6                  "host" : "bob.peer.net",
7                  "port" : 8080,
8                  "key" : "@ABCw1W19ZsKmG2KnfaoKIM66BRoreEkzaVm/J//w18=.ed25519"
9              }
10         },
11         -
12     }

```

Figure 2.2: Secure scuttlebutt protocol pub message format

The protocol operates under the assumption that each participant has an interest in only a portion of the entire data set. Participants choose their segment of the data pool by identifying the set of identities they are concerned with, regardless of the overall scale of the system. This guarantees that peers replicate only the data they consider significant based on identity [54].

2.2.2.2 Implementation

The Āhau application outlines three unique member roles in its design. These roles are defined by their responsibilities and access levels within the system, highlighting their separate duties.

- *Tribal Member* serves as the custodian of the record, indicating the initial source and any potential modifications.
- *Hapū Administrator* functions as the validator or supervisor, ensuring that the creators of the record comply with the general rules established for all members.
- *Iwi Administrator* acts as the record holder. All records created by tribal members, once moderated, are maintained by the Iwi administrator.

The application utilizing the scuttlebutt protocol and other methods comprises databases being backed up to servers managed by the tribe, and members managing their own distributed databases and cryptographic keys. Each subtribe will formulate and implement their own protocols, determining the types of information to be collected and setting various permissions for access and editing. They will also decide how this information will be shared among tribes, such as between whānau, hapū, and iwi, and manage the information accordingly [136]. The Āhau application is among the earliest solutions for indigenous data sovereignty, facilitating self-governance and control over the management of indigenous information, cultural preservation, and rights.

2.3 Rust Programming Language

The Rust programming language is a relatively new language that offers a unique set of features and capabilities to develop reliable and efficient systems [114]. Rust was introduced to address the shortcomings of existing system programming languages such as C and C++ [68], [121]. Rust incorporates modern programming language features while maintaining high performance and memory safety [18]. The origins of Rust can be traced back to the early 2000s, when a team at Mozilla began developing a new language to create a safe and concurrent language that would eliminate common programming errors such as null pointer dereferences, buffer overflows, and data races [81]. Motivations for building Rust include the need for a language that provides memory safety without sacrificing performance, as well as the desire to create a language that supports safe concurrent programming.

What sets Rust apart is its clear performance model, which enables developers to predict and reason about the efficiency of the program. This is achieved through fine-grained control over memory representations, including stack allocation and contiguous record storage [102]. The language strikes a balance between control and safety, ensuring the absence of data races, buffer overflows, stack overflows, and unauthorized memory accesses [75]. Rust is built on the following concepts that contribute to its safety and performance:

- **Ownership:** Rust uses an ownership model to manage memory. Each value in Rust has a variable called its owner. Only one owner can exist at a time, and when the owner goes out of scope, the value is automatically removed. This ownership model eliminates common issues, such as dangling pointers and memory leaks, making Rust programs safe by default [62].
- **Borrowing:** Rust's borrowing system allows for the temporary loaning of references to values, preventing data races, and enabling safe concurrent access to shared data. Rust's borrowing system is a crucial feature that allows for the temporary loaning of references to values. This ensures data safety by preventing data races and enables concurrent access to shared data without compromising its integrity [62], [65].
- **Lifetimes:** Rust's lifetime system provides a way to reason about the lifespan of references to ensure that borrowed references do not outlive the data to which they refer. This prevents the use

of references to invalid or deallocated memory, preventing crashes and other memory-related errors [62].

The combination of these features makes Rust a reliable and efficient choice for system programming. Developers have quickly embraced Rust for its ability to build high-performance applications without sacrificing safety. Furthermore, Rust's focus on zero-cost abstractions allows developers to write high-level code without sacrificing performance [68]. These abstractions are optimized at compile time, resulting in code that is as efficient as handwritten low-level code [68]. This makes Rust a powerful language for tasks that require both safety and performance, such as system programming, embedded systems, and networking applications.

Overall, Rust's emphasis on safety and performance has made it a standout language in the programming community. Its strong type system and ownership model ensure memory safety and eliminate common bugs that plague other systems programming languages such as C and C++. Rust's borrowing system allows for safe concurrent access to shared data, preventing data races, and ensuring the integrity of the program. Additionally, Rust's lifetime system helps to reason about the lifespan of references, avoiding crashes and memory errors.

2.4 Centralization

Centralization in the context of systems, software, and data involves the concentration of control and decision-making authority in a single entity or centralized authority [9] as seen in 2.3. This model has been the predominant approach in many industries and sectors, including data collection and management. Research findings have shown that large enterprises that adopt centralized control in data management are able to make more effective decisions, reduce costs, and provide better services to individuals [12], [14]. Since centralization is the act or process of bringing under a single control or authority, this approach can lead to a lack of transparency, accountability, and inclusivity, since decision-making power is concentrated in the hands of a few individuals or organizations [26].

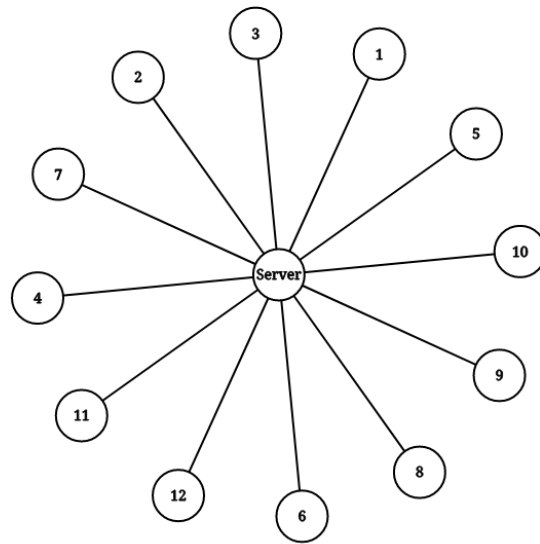


Figure 2.3: Client to server node placement

Centralization in the context of data governance has significant implications for indigenous communities and their pursuit of data sovereignty [24]. By centralizing control and decision-making power over data, indigenous peoples can be excluded from the process and have limited agency in determining how their data are collected, stored, and used. This lack of inclusion perpetuates a power imbalance and can lead to further exploitation and marginalization of indigenous communities. To overcome the challenges of centralization and promote the sovereignty of indigenous data, it is crucial to prioritize the principles of self-determination and autonomy [86], [95]. Indigenous communities should have the authority to determine the means of collecting, accessing, analyzing and distributing

data that align with their values, cultural protocols, and community needs [106].

2.4.1 Cloud Technology and Data Management

Cloud technology has become a crucial facilitator of centralized data management, providing adaptable, efficient solutions for storing, processing and retrieving large volumes of data [90]. The shift towards cloud computing is driven by its capacity to consolidate resources and allow users worldwide decentralized access. According to the authors [90], [117], cloud computing is a technology that enables broad, convenient, and immediate network access to a shared pool of flexible computing resources. This system has a notable impact on data management strategies, facilitating the shift from conventional on-premise data centers to cloud-based solutions. The authors highlight the importance of cloud services such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) in offering scalable and efficient data management solutions. These services support centralized data storage and processing, guaranteeing accessibility and reliability.

2.5 Decentralization

Rather than concentrating decision-making authority in one place, decentralization spreads it across multiple entities or nodes [97], as illustrated in 2.4. This approach can bring about a variety of advantages, such as improved transparency, robustness, and the ability of individuals or communities to retain control over their data [118]. Decentralization is based on the use of distributed ledger technologies, such as blockchain³, which allow peer-to-peer communication and data sharing without the need for intermediaries, such as centralized servers [105]. Decentralization as defined by the literature review involves the dispersal of power and decision-making authority, allowing individuals or communities to have greater agency and control over their data [78]. The dispersal of decision-making authority and control among multiple entities or nodes is known as decentralization. This approach provides greater transparency, robustness, and the ability of individuals or communities to have control over their data [73].

2.5.1 Peer-to-Peer Network

Peer-to-peer (P2P) networks are decentralized networks that allow direct communication and data sharing between individual devices, without the need for a central server [35], [52]. These networks are based on the collective resources and connectivity of all participants, making them resilient and scalable [11]. P2P networks have become increasingly popular in various applications, such as file sharing, messaging, and decentralized cryptocurrency systems [51]. In recent years, there has been a growing interest in harnessing P2P networks to share secure and private data.

³Blockchain and holochain are considered to be decentralized technologies with a different model in how they operate

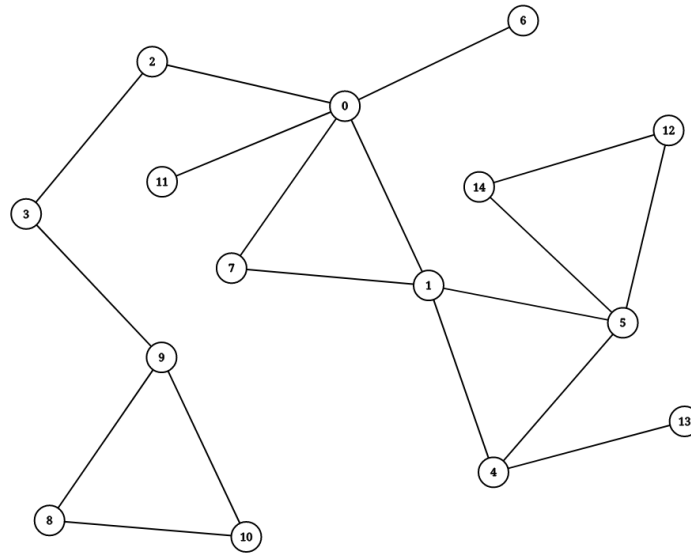


Figure 2.4: Decentralized clients

Peer-to-peer networks rely on the following pillars for their operation: direct communication, decentralization, resource sharing, intermediate brokers, and consensus mechanisms [115]. Direct communication allows nodes on the network to exchange information directly without relying on intermediaries. Decentralization ensures that there is no single point of failure or control, making the network more robust and resistant to censorship. Resource sharing allows participants to contribute their resources (such as computing power or storage) to the network, creating a distributed infrastructure [2], [115]. Consensus mechanisms ensure that all nodes in the network agree on the state of the system, enabling trustless interactions and preventing malicious activities.

However, it is important to note that peer-to-peer networks also face challenges. One of the main difficulties is the problem of finding distributed content within the network. As the size of the network increases, it becomes increasingly difficult to efficiently locate and retrieve specific data or resources [53], [91]. This problem has led to the development of various strategies and algorithms, such as distributed hash tables, content-addressable networks, and self-certifying identifiers, to facilitate content discovery in peer-to-peer networks [47]. These strategies aim to improve search efficiency and overcome the limitations of relying on centralized servers or indexes [47]. One notable example of a

peer-to-peer network that addresses the problem of content discovery is BitTorrent [42]. BitTorrent utilizes a peer selection strategy that combines reciprocity and the best experienced transmission rates, coupled with the election of the rarest fragments [13]. This approach ensures that users can efficiently locate and download content from multiple sources, maximizing the availability and speed of data retrieval.

Additionally, the scalability and heterogeneity of peers in a peer-to-peer network pose challenges for analytical modeling and optimization. Despite these challenges, peer-to-peer networks continue to evolve and thrive due to their inherent advantages [41]. Some popular applications of peer-to-peer networks include distributed content-sharing systems, large storage management systems, wide area name resolution services, and cooperative file systems [40]. These applications take advantage of the decentralized nature of peer-to-peer networks to create scalable and resilient infrastructures for data sharing, storage, and retrieval. In conclusion, peer-to-peer networks are important for their ability to provide a decentralized and resilient infrastructure for communication and data sharing [2], [5]. These networks are particularly relevant in today's digital landscape, where privacy, security, and censorship resistance are increasingly important considerations. Taking advantage of the power of direct communication, decentralization, resource sharing, and consensus mechanisms, peer-to-peer networks offer a compelling alternative to traditional client-server architectures. They eliminate single points of failure, distribute workload across the network, and provide alternate routing paths in case of link failure.

2.5.2 Distributed Hash Table

One of the key components of peer-to-peer systems is a DHT, which provides a standardized way to store and retrieve data in a decentralized manner. A distributed hash table is a fundamental component of peer-to-peer systems that enables efficient storage and retrieval of data in a decentralized manner [8], [43]. DHT requires a key-value pair, where the key is hashed to determine its location in the DHT. This allows efficient query routing and data retrieval in structured peer-to-peer networks [4], [43]. DHT relies on a decentralized and self-organizing network of nodes, where each node is responsible for a specific portion of the hash table. Various methods have been suggested to establish stable and consistent DHT peering. Some of these methods include Chord, CAN, Pastry, and Tapestry [15], [21]. For example, Chord is a protocol that uses consistent hashing to efficiently locate nodes in a network. By employing consistent hashing techniques, Chord ensures that each node is responsible for a specific section of the hash table based on its identifier [15], [21], [29].

Using consistent hashing, Chord enables efficient lookup and routing of data in the network, even when nodes join or leave dynamically. Super-peer networks, on the other hand, introduce a middle level into the network by designating certain nodes as super-peers. Super peers act as intermediaries between regular peers, facilitating efficient resource discovery and routing [7]. CAN is another distributed hash table protocol that divides the key space into a grid-like structure, with each node responsible for a specific region of keys [6]. Pastry, another distributed hash table protocol, uses a proximity-aware routing mechanism to ensure efficient lookup and retrieval of data. [6] In this approach, each node is

assigned an identifier based on a consistent hashing scheme, and the network is organized hierarchically. Tapestry, similar to CAN, uses a grid-like structure to divide the key space among nodes and provides efficient routing mechanisms for data retrieval.

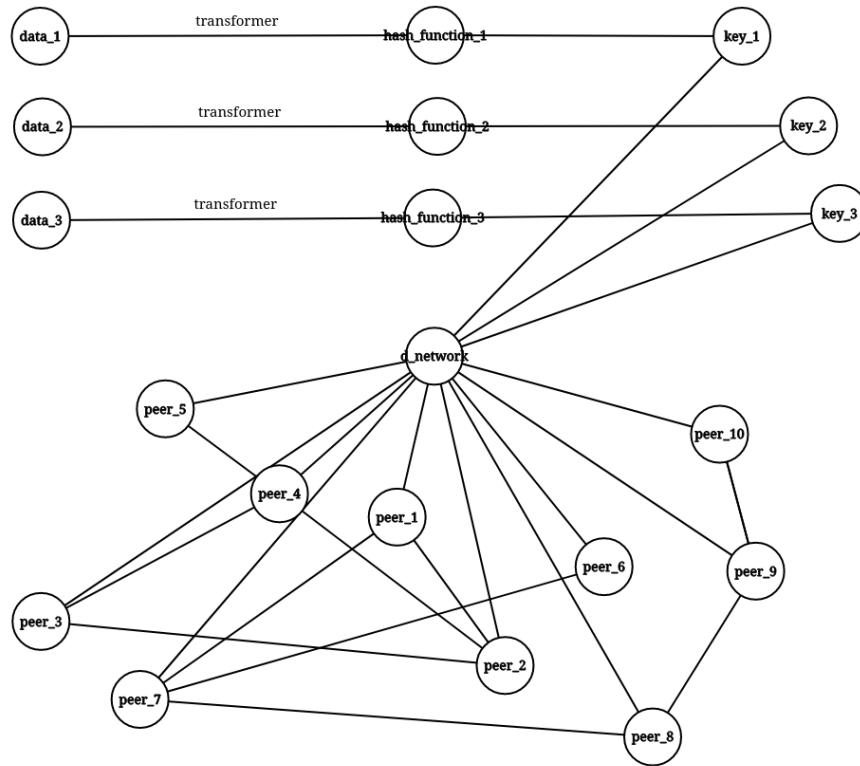


Figure 2.5: Data, hash function, key, peer network

These peer-to-peer protocols provide the infrastructure necessary for scalable distributed hash tables, allowing for efficient storage and retrieval of data in a decentralized manner. However, while distributed hash tables are efficient for data storage and retrieval, they face challenges when it comes to issues such as data consistency, fault tolerance, and security [43]. To address these challenges, researchers have proposed various techniques such as replication, consistency protocols,

and cryptographic mechanisms [44]. However, one of the limitations of these traditional distributed hash table protocols is their reliance on consistent hashing and a rigid structure that may not be suitable for certain applications or scenarios [108].

In summary, peer-to-peer networks utilize distributed hash table techniques, such as consistent hashing, to efficiently locate and route data in a decentralized manner. These techniques, such as Chord, CAN, Pastry, and Tapestry, provide scalability and fault tolerance in large-scale networks.

2.5.3 Blockchain

Blockchain technology, a Distributed Ledger Technology (DLT) is commonly perceived as a reliable and transparent solution to ensure the authenticity of data [60]. Through the use of cryptographic algorithms and dispersed consensus mechanisms, it guarantees the immutable and tamper-proof nature of data [37]. This makes blockchain a suitable technology to improve information security, confidentiality, and credibility in various contexts. One conceivable application of blockchain technology in the realm of data authenticity lies within cloud systems [109]. Using the transparency, traceability, and security attributes of the blockchain, cloud systems can increase the authenticity of data stored and managed within the cloud [45]. For example, blockchain can be used to generate an unalterable record of data modifications and transactions, allowing users to verify the authenticity and integrity of their data [109]. In addition, blockchain technology can also provide enhanced confidentiality and security features by eliminating the need for a centralized authority or intermediary. This can be particularly relevant in sectors such as healthcare, financial services, supply chain management, and identity management, where data authenticity and security constitute crucial considerations [22], [39], [45].

The blockchain relies on the following concepts and features that contribute to its security and trustworthiness.

1. **Decentralization:** The blockchain operates on a decentralized network of nodes, where data are stored and verified by multiple participants. This decentralization ensures that no single entity has control over the entire blockchain network, making it more resistant to attacks and manipulation [37].
2. **Consensus Mechanisms:** Blockchain uses consensus mechanisms, such as Proof of Work or Proof of Stake, to validate and agree on the transactions and data added to the blockchain. These mechanisms ensure that any changes or additions to the blockchain must be agreed upon by the majority of participants, preventing fraudulent or unauthorized modifications [37].
3. **Data Encryption and Hashing:** The blockchain uses cryptographic algorithms to ensure secure data transfer and storage. This includes encrypting data to protect its confidentiality and integrity, as well as using hashing algorithms to create unique identifiers for each block of data. These identifiers, known as hashes, are used to link blocks in a chain-like structure, ensuring the immutability and tamper-proof nature of the blockchain [46].

4. **Smart Contracts:** Smart contracts are self-executing contracts with predefined rules and conditions encoded in the blockchain. These contracts automate and facilitate the execution of transactions, eliminating the need for intermediaries and reducing the risk of fraud or manipulation [37], [46].

However, while blockchain technology has its advantages in terms of data integrity and security, it also has certain limitations. For example, blockchain technology is often associated with high energy consumption and scalability issues [46]. The mining process and reaching consensus on a blockchain network can be time consuming and resource intensive, making it challenging to handle large volumes of data promptly [37]. Furthermore, blockchain technology can also present challenges in terms of data privacy. Blockchain transparency, while beneficial to data integrity, can also expose sensitive information to unauthorized parties [72]. Furthermore, the integration of blockchain technology with existing cloud systems can be complex and may require significant changes in infrastructure and processes.

In conclusion, blockchain technology offers promising solutions to improve data integrity and security in various industries. Its transparent and tamper-proof nature can assure users of the authenticity and integrity of their data [37]. However, it is important to consider the limitations of blockchain technology, such as its energy consumption and scalability challenges [37]. Additionally, the integration of blockchains with existing systems may require careful planning and infrastructure modifications. Despite these challenges, the potential benefits of blockchain in terms of data integrity and security cannot be ignored. As technology continues to evolve, solutions to these limitations are expected to be developed, making blockchain an even more viable option to protect customer data and ensure data integrity in the future. In general, blockchain technology has shown great potential to improve data integrity and security in various industries.

2.5.4 Holochain

Holochain is an emerging technology that offers a decentralized approach to data management and empowers people with greater control over their own information [105]. Unlike centralized systems, holochain allows users to store and manage their data locally on their own devices through a framework called agent-centricity [100], [124]. This unique architecture enables each user to have an independent chain, known as "DNA," which holds and verifies their data and interactions [124]. By distributing processing tasks among network nodes, holochain achieves high scalability and improves efficiency compared to traditional blockchain networks, resulting in faster transaction speeds.

It is important to note that the holochain platform is in its early stages of development and is not yet widely adopted. Therefore, the existing literature on holochain is limited, and more research and experimentation is needed to fully understand its capabilities and limitations. Despite its early stage of development, holochain holds promise as a decentralized data management solution.

2.5.4.1 Ceptr

The holochain project is closely related to the Ceptr project, which is its parent project. Ceptr is a meta-framework that provides the underlying design principles and patterns for holochain [25], [99], [139]. Through Ceptr, holochain inherits the concept of "interoperable patterns" that allow for modular development and easy integration of different applications [99]. The combination of holochain's agent-centric architecture, mutual credit validation process, and integration with Ceptr [32] makes it a promising technology for supporting IDS.

2.5.4.2 Core Pillars

Holochain contains many fundamental pillars to the underlying architecture, two in particular:

- **Intrinsic Data Validity:**
In holochain, data are structured so that its integrity is embedded within itself, rather than relying on external access controls or firewalls. This ensures that the data remain tamper-proof and trustworthy, as any attempt to alter or manipulate the data would break the integrity of the chain [76], [109].
- **Peer Witnessing:**
Holochain utilizes a concept called "peer witnessing" to validate and verify the data stored on each user's chain. This involves peers in the network acting as witnesses to each other's interactions, ensuring the accuracy and authenticity of the data [71], [76].

These two core concepts of holochain, intrinsic data validity, and peer witnessing enable the technology to provide scalability and fault tolerance in large-scale networks [70]. This distinguishes holochain from centralized architectures and traditional blockchain systems, where all nodes in the network must reach a consensus on each transaction, leading to scalability challenges and latency in processing transactions.

2.5.4.3 Architecture

The holochain, which is part of the larger Ceptr project, takes influence from the sciences that involve the study of complex adaptive systems, such as biology [139]. This influence is embedded in the design of holochain's underlying architecture, which introduces key terms and concepts related to its functioning, for example:

Holochain Application holochain applications consist of a client and a holochain "backend", often referred to as an "hApp". The hApp is a collection of "DNA's" or the equivalent of microservices, which too are a collection of Zome's (chromosomes) as defined in the context of holochain [25], [124]. Zomes can be seen as modular units or executable code that encapsulate specific functionalities and can interact with each other through defined interfaces. To better understand this architecture, let us delve into the specific terms and their meanings.

Biological System	Holochain System
Cells	Cells
DNA	DNA
Chromozomes	Zomes
Organism	Agents

Table 2.2: Comparison of biological and holochain systems naming convention

The innermost module is a Zome Function. The Zome is a cohesive unit of executable code that exposes some functionality and data structures as an API. There are two types of Zome, that is, integrity Zome, which defines data types, and coordinator Zome, which defines the application logic of that particular hApp [59], [124]. One or more zomes together make up a DNA, which is equivalent to a service or a microservice in traditional terms. When a DNA is active or "comes alive", it is bound to an agent ID which is the unique identifier of the participant; this is referred to as a cell [59], [124]. In other words, a cell is created when an ID of the agent is bound to active DNA.

There can be many cells within an hApp, each cell can be slotted into different roles within a hApp which form the application's back-end. Each participant has a conductor who is responsible for managing the cells within the hApp, including creating and joining networks, validating transactions, and maintaining the participant's source chain [105], [124]. The conductor is the application server runtime that runs on the participant's device and handles all interactions with the holochain network. The conductor can be equivalent to a server in traditional client-server architectures [105]. It is both a client and a server for other peers within the network. The agent is the individual participant or entity within the holochain network.

In summary, holochain is a framework that draws inspiration from natural sciences and biology. It operates on the concept of hApps consisting of zomes that form a DNA structure. These DNAs, when activated, are associated with agent IDs and generate cells within the hApp. The conductor manages these cells, acting as the back-end of the application. This unique architecture allows individual participants or agents to maintain control over their data and interactions in the network.

Chapter 3

Methods and Methodology

This section provides an overview of the methods and methodology used in this investigation. To explore whether holochain technology can be used for indigenous data sovereignty, a mixed-method approach was adopted. The research began with a literature review to establish a solid understanding of holochain technology and its potential applications in the context of indigenous data sovereignty. The literature review involved an in-depth analysis of academic papers, reports, and case studies related to holochain technology and indigenous data sovereignty. A review of the literature was also conducted to assess the strengths and weaknesses of previous studies and identify knowledge gaps. After reviewing the literature, an experiment was conducted to assess the feasibility of using holochain technology for indigenous data sovereignty. The experiment involved developing a prototype holochain application that simulated data sharing and governance within an indigenous community.

This thesis aims to critically evaluate whether holochain technology can effectively support indigenous data sovereignty. To achieve this, the research first explores the literature on holochain technology, examining its underlying architecture, features, and potential applications. This basis is used to create context around phases of the experiment conducted to assess the feasibility and effectiveness of holochain technology for indigenous data sovereignty. There are three core phases to this research.

- Build an infrastructure that mimics a centralized platform where data is shared between different services and systems according to industry standards.
- Design and develop a prototype holochain application that enables data transmission between peers within a network.
- Attempt to share data across a peer network to a traditional platform with each peer having the ability to remove their data at will.
- The results and challenges of each phase and their limitations.

The chosen methodology is Design Science Research, which is a problem-solving approach that focuses on creating innovative solutions to real-world problems. This approach involves iterative cycles of design, development, and evaluation to iteratively improve and refine the solution.

3.1 Design Science Research

DSR differs from traditional research methodologies, such as positivist or interpretive approaches, in several key ways. First, the DSR is problem-driven rather than theory-driven. The primary goal of DSR is to develop practical solutions to real-world problems, while traditional research methodologies focus on theoretical understanding and explanation [48], [96], [111]. Second, DSR involves iterative cycles of design, development, and evaluation. This iterative approach allows for continuous refinement and improvement of artifacts based on feedback and evaluation results. In addition, the DSR emphasizes the importance of practical relevance and applicability [10].

Unlike other research paradigms that focus primarily on understanding and explaining phenomena, DSR aims to create practical solutions that serve human purposes. Some [66] argue that there is a significant overlap between DSR and action research methodologies. Both prioritize problem solving and focus on practical outcomes. However, DSR differs from action research in its emphasis on artifact creation and evaluation of artifacts [55]. Although action research focuses on the active participation of researchers in solving practical problems within organizational contexts, DSR places a central focus on the design and the proof of the usefulness of the created artifact [55].

Design Science Research Methodology consists of several key components that make it effective in Software Engineering research. The key components of design science research methods include the following:

1. Identification and motivation of problems: This is the first step in Design Science Research, where researchers identify a specific problem or challenge in software engineering and provide a justification for the need to solve it [3].
2. Artifact Design: In this phase, researchers create a conceptual design of the artifact that will serve as a solution to the identified problem.
3. Artifact Construction: Once the design is finalized, researchers proceed to build and develop the artifact according to the defined design.
4. Validation: After the artifact is built, validation is performed to ensure that it meets the requirements and objectives set out in the problem identification phase. The validation phase involves testing and refining the artifact to ensure its effectiveness and usability.
5. Evaluation: This phase involves assessing the impact and effectiveness of the artifact in solving the problem. It includes analyzing the performance of the artifact through the use of indicators and data obtained from practitioners who have used the artifact in real world situations.

6. **Communication:** The final step in Design Science Research Methodology is the communication of the research findings and the artifact itself. Researchers present their findings and artifact to relevant stakeholders, such as researchers, practitioners, and the wider scientific community. This step ensures that the knowledge and insights gained from Design Science Research are shared and accessible to others.

These components make Design Science Research Methodology an effective approach in Software Engineering, as it provides a structured and systematic process to address complex problems and develop innovative solutions that are relevant and impactful in the field [19], [96].

3.2 Artifacts

In design science research methods, artifacts refer to tangible outputs or solutions that are created to address a specific problem. These artifacts can take various forms, such as software prototypes, frameworks, models, algorithms, or any other designed object that offers a solution to the research problem [101], [111]. These artifacts are designed and developed based on the conceptual design and specifications defined in the problem identification phase [19]. Artifacts play a crucial role in design science research methodology, as they are tangible outcomes that demonstrate the effectiveness and feasibility of the proposed solution. For example, in the context of indigenous data sovereignty, an artifact could be a software application or platform that allows indigenous communities to securely store and manage their data, ensuring their control and ownership over those data.

3.3 Theory versus Action

In the context of design science research methods, theory refers to conceptual frameworks, principles, and knowledge that inform the design and development of artifacts [17]. Theory guides the decision-making process and provides a basis for understanding the problem and potential solutions [34]. On the other hand, action refers to the iterative process of designing, developing and evaluating artifacts experimentally [92]. The actions taken during the development and evaluation of artifacts in Design Science Research Methodology are based on theoretical principles and knowledge, but are implemented through practical applications [111]. For example, in the context of indigenous data sovereignty, theoretical knowledge about data ownership and governance principles of indigenous communities would inform the design of an artifact that empowers indigenous communities to have control over their data.

3.4 Evaluation

Evaluation plays a crucial role in science research methods in design, as it assesses the effectiveness, feasibility, and value of the artifacts developed. Evaluation in design science research methods involves

assessing artifacts against predefined objectives and criteria. This evaluation is typically carried out through rigorous testing, experimentation, and user feedback [66], [88]. Some common evaluation methods used in Design Science Research Methodology include:

- **Observational case study:** This method involves studying the artifact in depth within a real-world business environment to understand its performance and impact [19].
- **Field study:** This method involves monitoring the use of the artifact in multiple environments to gather data on its effectiveness and usability [19], [92].
- **Experimental study:** This method involves conducting controlled experiments to compare the performance of the artifact against alternative solutions or benchmarks [34].
- **Surveys and Interviews:** This method involves gathering feedback from stakeholders, users, and experts through surveys and interviews to understand their perspectives on the artifact's efficiency and usability [19], [92], [101].
- **User Feedback:** This method involves collecting feedback from users who have interacted with the artifact to understand their satisfaction levels, identify areas for improvement, and gather insights into the artifact's usability [19].
- **Expert Review:** This method involves expert evaluators assessing the artifact's design, functionality, and adherence to best practices and standards [10], [67].

Moreover, in the context of indigenous data sovereignty, evaluation should also consider the impact of the artifact on indigenous communities' data governance and control. This can be done by engaging indigenous communities in the evaluation process and collecting feedback on whether the artifact aligns with their values, needs, and aspirations.

In the context of empowering indigenous communities to have control over their data, the design of an artifact should consider the following key aspects. *Respect for Indigenous Knowledge Systems:* The artifact should be designed in a way that respects and incorporates Indigenous knowledge systems, values, and practices. *Data Ownership and Control:* The artifact should allow indigenous communities to have ownership and control over their data, including the ability to determine who can access and use their data and for what purposes. *Equitable Benefit Sharing:* The artifact should promote equitable benefit sharing, ensuring that indigenous communities receive fair benefits from the use of their data and that the benefits are not solely accrued by external entities.

Chapter 4

Research Phase 1: Infrastructure Build and Design

The first phase of research is focused on developing an infrastructure prototype and designing the artifact for the sovereignty of indigenous data. This phase involves creating an infrastructure that, by modern industry standards, should satisfy a centralized platform that stores and manages data. Although this stage is inherently characteristic of contemporary infrastructure, its objective also includes emulation of potential challenges that may arise from modern infrastructure practices. This is in response to the need for a system redesign that takes into account the principles of indigenous data sovereignty.

4.1 Goals and Objectives

The research phase 1 aims to achieve the following goals and objectives.

- Design an infrastructure prototype that can serve as a foundation for the artifact.
- Develop the artifact with relevant features that promote data sharing across different services, such as a database, a user interface, and communication protocols.
- To develop the artifact with features that promote systematic data governance and control, data ownership by the platform with sharing capabilities.

In the development of this specific stage, we made sure to provide a comprehensive overview of the sequential procedure and the thought process involved in the construction of the infrastructure. This was done in part to keep up with the rapidly evolving changes in the industry, as new advancements and updates in technology are occurring worldwide.

4.2 Defining Requirements

To successfully achieve the goals and objectives of this research phase, it is essential to define the key requirements of the infrastructure prototype and artifact. Identifying specific requirements will help guide the design and development process. As mentioned in the literature review, AWS provided countless services that can be used to create the infrastructure for the prototype. Since AWS is widely used and provides a comprehensive set of tools and services, it is a suitable choice to develop the infrastructure prototype. Some of the key requirements for the infrastructure prototype and artifact include the following.

- **Scalability:** The infrastructure prototype should be able to handle a large volume of data and users, allowing future growth and expansion.
- **Interoperability:** The infrastructure should be able to integrate and communicate with other systems and platforms, allowing seamless data sharing and collaboration.
- **Security:** The infrastructure prototype should have robust security measures in place to protect the data and ensure privacy.
- **Data sovereignty:** The infrastructure prototype should provide mechanisms for indigenous communities to have ownership and control over their data, including the ability to determine who can access and use their data.

These requirements are critical because they must ensure that best practices are followed in terms of scalability, interoperability, security, and data sovereignty. This is crucial within modern-day development methodologies, especially when dealing with sensitive data such as indigenous data. Based on the identified requirements, it is clear that the choice of infrastructure for the prototype must support scalability, interoperability, security, and data sovereignty to meet these requirements, and the infrastructure prototype should leverage cloud services. With numerous top cloud computing services available, we opted for Amazon Web Services because of its incorporation of various development tools linked to this cloud service, one of which will be utilized in this phase. The AWS services we plan to use will be determined by the prototype artifacts in alignment with the goals of this phase.

- AWS API Gateway
- AWS Cloudwatch
- AWS Cognito
- AWS DynamoDB
- AWS IAM

- AWS Lambda
- AWS RDS
- AWS S3

Given the requirements mentioned, a project structure 4.1 has been defined to represent the infrastructure being built.

The project structure directories contain separate instances of stacks, configurations, and services, which are independent of each other. The package directory includes core functions that are primarily used to handle the defined services within our stacks. Each subdirectory in the functions directory corresponds to specific handlers and services based on the naming convention relevant to the infrastructure stack and the construction that is being performed. Name conventions provide an indication and representation of possible services that could be potential in nature and do not fully represent the service. SST offers an effective solution to set up the infrastructure in a manner similar to a typical JavaScript project, incorporating contemporary methodologies.

The following 5.4 is a display of how identically similar the call scripts would be to that of JavaScript compared to the likes of Terraform or AWS cloud formation that are most common in platform engineering practices.

4.2.1 Controlling Access to Data

The AWS services that will be used in the infrastructure prototype can help address the requirement of controlling access to data by implementing proper authentication and authorization mechanisms. For authentication, AWS Cognito can be used as an intermediary service to handle user sign-up, sign-in, and account management [87]. Provides user authentication and registration services that allow users to securely access the system. On the other hand, for authorization, AWS IAM can be utilized [58]. It allows administrators to define granular permissions and access control policies for users, ensuring that only authorized individuals can access and manipulate data.

4.2.2 Scalability

The infrastructure prototype should be able to handle a large volume of data and users, allowing future growth and expansion. To address this requirement, the infrastructure prototype can leverage AWS services such as AWS Lambda and AWS DynamoDB. AWS Lambda allows the infrastructure to automatically scale on the basis of the incoming request load, ensuring that resources are allocated efficiently [36]. AWS DynamoDB, on the other hand, is a highly scalable and flexible NoSQL database service that can handle millions of requests per second. It provides automatic scaling and can handle large amounts of data, ensuring that the infrastructure prototype can accommodate scalability requirements.

4.2.3 Interoperability

Interoperability is crucial for the infrastructure prototype, as it requires seamless integration with various systems and technologies. To achieve interoperability, the infrastructure prototype can make use of the AWS API Gateway. AWS API Gateway acts as a front door for applications to access data, business logic, or functionality from back-end services. Provides a fully managed service to create, deploy, and manage APIs [64]. With AWS API Gateway, the infrastructure prototype can create RESTful APIs that can be easily consumed by different systems and technologies, enabling seamless interoperability. Web sockets can also be utilized for real-time communication and interoperability between different components of the infrastructure prototype and potentially external systems such as holochain applications. In addition to the AWS API Gateway, AWS AppSync can also be used to achieve interoperability within the infrastructure prototype. AWS AppSync is a fully managed GraphQL service that simplifies application development by enabling real-time data synchronization and offline capabilities. However, AWS AppSync would not be used in this case, as the infrastructure prototype will not require the features provided by GraphQL.

4.2.4 Security and Privacy

Security and privacy are paramount considerations for the infrastructure prototype. To ensure the security and privacy of the infrastructure prototype, AWS provides several services and features that can be leveraged. Some of these services and features include:

- AWS Identity and Access Management: IAM allows users, groups, and roles to be created with specific permissions to access AWS resources [58].

The service provides granular control over who can access the infrastructure prototype and what actions they can take. We will be using AWS IAM to manage user authentication and authorization to access the infrastructure prototype and create development and production environments for appropriate access control.

4.2.5 Data Storage

AWS provides a variety of data storage options to meet the needs of the infrastructure prototype. For this purpose, we will use some of the following AWS storage services.

- Amazon S3: Amazon Simple Storage Service provides highly scalable object storage to store and retrieve any amount of data. The infrastructure prototype will use Amazon S3 to store and retrieve data securely and reliably [36].
- Amazon RDS: Amazon's Relational Database Service offers managed relational databases in the cloud. We will utilize Amazon RDS for the infrastructure prototype to store and manage structured data securely, which may come from holochain if possible [36].

- Amazon DynamoDB: Amazon’s fully managed NoSQL database service. Provides fast and flexible storage for structured and semi-structured data [36].

4.3 Prototype Design and Implementation

When designing and implementing the infrastructure prototype, it is important to consider best practices and use AWS services and features that support the scalability, resilience, and high availability of the system. Although these services can be launched through manual provisioning, we will leverage infrastructure-as-code approaches to automate the deployment and management of our infrastructure prototype.

- AWS CloudFormation: AWS CloudFormation allows for the provisioning and management of AWS resources using a declarative template. With AWS CloudFormation, we can define our infrastructure in a template file and use it to create or update resources in a consistent and repeatable manner across multiple environments [64].

However, AWS CloudFormation can be difficult to manage when it comes to complex infrastructure deployments. To address this challenge, we can utilize AWS CloudFormation StackSets with external tools such as AWS Cloud Development Kit, Terraform, or SST. Using these tools, we can define our infrastructure as code and easily manage the deployment and updates of resources across multiple AWS accounts and regions. The decision to choose SST as the tool to manage AWS CloudFormation StackSets was made due to its simplicity and ease of use, especially for developers with limited infrastructure management experience. This allows for a more efficient and streamlined deployment process, reducing the potential for manual errors and ensuring consistent infrastructure setups across different environments.

4.3.1 SST

SST is an open-source framework for building serverless applications on AWS. SST provides an intuitive and developer-friendly experience for defining infrastructure as code, deploying serverless resources, and managing application configurations [119]. SST uses typescript as its primary programming language and leverages AWS CloudFormation under the hood to provision and manage resources. The use of SST in the design and implementation of infrastructure prototypes provides several benefits. Some of the benefits of using SST in the infrastructure prototype design and implementation are as follows:

- Simplified Infrastructure Management: SST abstracts the complexities of managing AWS CloudFormation resources and provides a simplified and intuitive interface to define and deploy infrastructure components.

- **Improved Developer Experience:** SST utilizes TypeScript as its primary programming language, offering developers a familiar and productive environment to build serverless applications.
- **Efficient Deployment Process:** SST automates the deployment process, allowing developers to easily deploy and update their serverless applications with minimal effort. This results in faster development iterations and reduces the potential for manual errors.
- **Enforced Infrastructure Consistency:** SST ensures consistent infrastructure setups across different environments by leveraging AWS CloudFormation under the hood.
- **Easy Collaboration:** SST enables easy collaboration among team members by providing a shared infrastructure configuration and deployment workflow. This allows teams to work seamlessly together and ensures that everyone has access to the same infrastructure setup.

Opting for SST offers the advantage of streamlining system design and infrastructure management by enhancing usability and simplicity while still ensuring practical control over deployments and infrastructure components. It was crucial to have a comprehensive understanding of how infrastructure is deployed, used, and recorded during this specific phase. This understanding was based on a version control system that also recognized the underlying process, avoiding dependence on a single vendor, and integrating well with other systems to also replicate real-world situations.

4.3.1.1 API Stack

The API stack is a key component of the design and implementation of the infrastructure prototype. It serves as the interface between the serverless application and external systems or clients. In adherence to the recommended guidelines, we implemented conventional authentication and integrated the SNS stack to manage topics. We have designated each route to require JWT authentication, except for public handlers or routes which do not necessitate authentication. By clearly defining this approach in our procedure, we guarantee that the stack generates a meaningful output that showcases important details of the logged events in both AWS and our custom infrastructure. To incorporate the API stack into the SST, the procedure 1 was followed using the following implementation approach.

Procedure 1: API stack

Definition:

Assume that the stacks listed in the requirements are subsequent stacks already established and are being imported. Each stack in the requirements are AWS resources. All requirements are imports of the SST library. Assume r_{public} to be the number of public routes and $r_{private}$ to be the number of private routes internally.

Require: Api, AuthStack, SNSStack, use, StackContext

Result : auth

```

1 auth ← use(AuthStack)
2 topic ← use(SNSStack)
3           ▷ Property of API should be set to reflect modern authentication practices
4 api ← authorizer ← default ← [jwt]
5           ▷ Routes are to be defined appropriately
6 public_route ← (r_public = 2), private_route ← (r_private = 1)
7 hApp_handler, hAppAgent_handler ← lambda_function
8           ▷ The following routes are placed here to display public and internal routing
9 for each handler to i do
10 |   POST ← public ← hApp_handler
11 |   GET  ← public ← hApp_handler
12 |   GET  ← private ← hApp_handler
13 end
14           ▷ Ensure the stack returns an output that is useful to the developer log
15           ▷ HTTP api properties
16 ApiID ← api.id
17 ApiEndpoint ← api.url
18 ApiAvailableRoutes ← api.routes
19 return api

```

The design of our API stack includes the integration of handlers that enable external HTTP calls from a client to the created stack. These handlers, as a result of the design decisions made by SST, provide developers with the ability to make live code edits, with event listeners monitoring any changes made to the project once it is running. Despite being modified locally, these changes allow us to observe updates on the AWS platform within seconds. By implementing simple API calls, we were able to test both public and private routes when running the full stack. Once the stack is running, it generates a useful log that allows the developer to access information about the available routes and other necessary details related to the stack that has been created. The handlers have the capability to be expanded in real-time to accommodate observable modifications, which can prove highly beneficial for debugging and delivery purposes.

4.3.1.2 Authentication Stack

The auth stack is of utmost importance in deciding the data access of applicants based on their role and permissions. This implementation is evident in SST, as depicted in 2. The design choice is founded on the principle that the majority of contemporary or obsolete systems incorporate some form of email and password authentication. In this process, we explicitly specify the service to be utilized within AWS as cognito, with the necessary stack context where the essential attribute is an email. In accordance with the specified service, we guarantee that the procedures generate a valuable log that allows us to identify the constructed elements and the necessary user pool data for troubleshooting. Additionally, we retrieve the region and application stage to compare the development stage with the production stage, which is a common practice in the industry.

Procedure 2: Authentication stack

Definition:

Assume that the stacks listed in the requirements are subsequent stacks already established and are imported. Each stack in the requirements are AWS resources. All requirements are imports of the SST library.

Require: use, Cognito, StackContext

Result : auth

```

1                                     ▷ Get the cognito stackcontext to be stored in the auth
2 auth ← Cognito(stack)
3                                     ▷ Ensure the properties being defined for a login requires an email
4 auth ← login ← [email]
5                                     ▷ Ensure the stack returns an output that is useful to the developer log
6                                     ▷ Staging properties
7 Region ← app.region
8 AppStage ← app.stage
9                                     ▷ Authentication properties
10 UserPoolId ← auth.userPoolId
11 UserPoolArn ← auth.userPoolArn
12 UserPoolClientId ← auth.userPoolClientId
13 CognitoIdentityPoolId ← auth.cognitoIdentityPoolId
14 return auth

```

After constructing the authentication stack, we guarantee to provide the stackcontext to any other service that wishes to use the authentication service for accessing a specific resource.

4.3.1.3 Database Stack

The database stack is where we specify our database requirements based on our assumed future anticipated interactions with holochain. This is illustrated in procedure 3, which showcases the specific database we have selected and specifies its engine type. Furthermore, we implement two NoSQL tables to store information about the holochain app services that interact with the agent connections, as well as the holochain agent records that establish connections with the cloud platform. In both scenarios, we collect and document events to provide evidence that the services of agents connecting to the cloud are recorded securely and the connections are successfully established.

Procedure 3: Database stack

Definition:

Assume that the stacks listed in the requirements are subsequent stacks already established and are imported. Each stack in the requirements are AWS resources. All requirements are imports of the SST library.

Require: use, RDS, Table, StackContext

Result : database, hAppAgentServices, hAppAgentConnections

```

1                                     ▷ Get the database and table context to be instantiated
2 database ← RDS(stack)
3 hAppAgentServices ← Table(stack)
4 hAppAgentConnections ← Table(stack)
5                                     ▷ Ensure the properties being defined for the database engine is current
6 engine ← "postgresql 13.9" || "mysql.x"
7     ▷ Ensure the properties being defined for the hAppAgentServices defines a primary index
8 primaryIndex ← {partitionKey ← id, sortkey ← name}
9                                     ▷ Ensure the stack returns an output that is useful to the developer log
10                                     ▷ Database cluster properties
11 DatabaseId ← database.id
12 DatabaseName ← database.defaultDatabaseName
13 DatabaseClusterArn ← database.clusterArn
14 DatabaseClusterPort ← database.clusterEndpoint.port ← string
15 DatabaseClusterHostname ← database.clusterEndpoint.hostname
16                                     ▷ DynamoDB properties
17 DynamoDbHolochainAppAgentServices ← hAppAgentServices.tableName
18 DynamoDbHolochainAppAgentConnections ← hAppAgentConnections.tableName
19 return database, hAppAgentServices, hAppAgentConnections

```

Once the database stack is built, we guarantee the provision of a developer log that provides detailed information about the database properties, including the cluster ARN, as well as the creation of two NoSQL tables for agent services and connections. Furthermore, we guarantee the return of the

database stack as a resource, as well as the agent service and connections resource, which may prove useful for any other internal service we may utilize.

4.3.1.4 SNS Stack

The SNS stack is where we specify topics that services can subscribe to or integrate with automatic scaling as described in 4. This implies that the infrastructure can handle load increases without requiring manual intervention, making it well suited for applications with fluctuating traffic. In a serverless environment, we are only charged for the resources that we utilize. Using the SST framework alongside AWS services such as Lambda and SNS, we are billed solely for the compute time consumed, thereby reducing the expenses associated with maintaining idle servers. The pub/sub pattern is highly effective in the creation of loosely coupled systems. Messages are published on a topic by services without any knowledge of the subscribers, who can independently process the received messages. This allocation of duties leads to a system that is more robust and simpler to maintain. AWS services like SNS are specifically engineered to provide high availability and durability. By integrating these services into our infrastructure, we can guarantee that our application can consistently process and transmit messages. With AWS's shared responsibility model, security is a joint concern between AWS and the developers of the application. AWS takes care of cloud security, while SST assists in ensuring secure application development by managing various security best practices.

Procedure 4: SNS stack

Definition:

Assume that the stacks listed in the requirements are subsequent stacks already established and are imported. Each stack in the requirements are AWS resources. All requirements are imports of the SST library.

Require: use, Topic, DatabaseStack, SQSStack, StackContext

Result : topic

```

1                                     ▷ Set up dynamo and queue for a holochain service
2 database ← RDS(stack)
3 hAppAgentServices ← use(DatabaseStack)
4 whanauRecordQueue ← use(SQSStack)
5 analyticsQueue ← use(SQSStack)
6 researchQueue ← use(SQSStack)
7                                     ▷ Ensure a topic is linked to the Holochain Agent Service connector
8 topic ← Topic(stack)
9 for each default property in topic to i do
10 |   timeout ← 30 seconds
11 |   environment ← {tableName ← hAppAgentServices.tableName}
12 |   permissions ← [hAppAgentServices]
13 |   bind ← {whanauRecordQueue, analyticsQueue, researchQueue}
14 end
15                                     ▷ Link the subscribing handler to the topic
16 subscribers ← holochainIntegratedServiceQueue[handler]
17                                     ▷ Ensure the stack returns an output that is useful to the developer log
18                                     ▷ Topic properties
19 TopicId ← topic.id
20 TopicName ← topic.topicName
21 TopicSubscriber ← topic.subscriptions[string]
22 return topic

```

The determination of which subjects to publish is based on the establishment of a connection via an incoming web socket from a holochain application. Ultimately, the creation of a serverless infrastructure that utilizes SST and AWS services such as SNS for a pub/sub system delivers a scalable, cost-effective and reliable resolution. This approach supports a decoupled architecture, guaranteeing the sustainment and strength that are essential for a well-crafted infrastructure.

4.3.1.5 SQS stack

In the domain of serverless architectures, it is crucial to incorporate message queues and stream processing services to effectively manage asynchronous tasks and guarantee reliable data processing

pipelines. The Serverless Stack (SST) framework offers features for both Amazon Simple Queue Service (SQS) and Amazon Kinesis Data Streams, which serve different purposes and provide distinct capabilities.

Amazon SQS is a message queuing service that is fully managed. It allows for the decoupling and scaling of microservices, distributed systems, and serverless applications [58]. In the SST framework, the queue construct is used to create an SQS queue. This construct allows for the creation of a consumer Lambda function that is triggered to process messages from the queue. The main benefit of using SQS is its simplicity and the assurance of message delivery. However, it is important to mention that SQS only allows one consumer per message [58], which makes it a suitable option for tasks that involve straightforward, one-to-one communication between components.

On the contrary, Amazon Kinesis Data Streams is specifically designed for streaming and processing real-time data. The Kinesis stream construct in SST simplifies the creation of a Kinesis stream, which is capable of efficiently handling a large volume of data. Unlike SQS, Kinesis enables multiple consumers to process data simultaneously, facilitating parallel processing of data by different components. This functionality is especially advantageous for intricate distributed applications that require real-time data analysis and processing. Additionally, Kinesis Data Streams retains data records for a default duration of 24 hours, which can be extended up to 365 days. This allows data reprocessing and guarantees data durability.

To summarize, the decision between SQS and Kinesis Data Streams in the SST framework should be based on the particular needs of the application. For simple point-to-point messaging patterns, SQS provides a direct and reliable solution. However, Kinesis Data Streams is suitable for applications that demand real-time data processing with multiple consumers. It offers a strong infrastructure that can handle large-scale data streams. The inclusion of these services in serverless architectures highlights the dedication to constructing applications that are scalable, efficient, and resilient. In this section, we establish the guidelines for generating SNS topics for a queue in the event of multiple incoming connections from agents. These guidelines are described in procedure 5. We chose SQS due to its simplicity, reproducibility, and potential for future extensions if modifications are required.

Procedure 5: SQS stack

Definition:

Assume that the stacks listed in the requirements are subsequent stacks already established and are imported. Each stack in the requirements are AWS resources. All requirements are imports of the SST library.

Require: Queue, StackContext

Result : whanauRecordQueue, analyticsQueue, researchQueue

```

1                                     ▷ Build appropriate queues within SQS service
2 whanauRecordQueue ← Queue ← whanauRecord_handler
3 analyticsQueue ← Queue ← analytics_handler
4 researchQueue ← Queue ← research_handler
5                                     ▷ Ensure the stack returns an output that is useful to the developer log
6                                     ▷ Whanau properties
7 WhanauDbQueueId ← whanauRecordQueue.id
8 WhanauDbQueueName ← whanauRecordQueue.queueName
9 WhanauDbQueueUrl ← whanauRecordQueue.queueUrl
10                                     ▷ Analytics properties
11 AnalyticsQueueId ← analyticsQueue.id
12 AnalyticsQueueName ← analyticsQueue.queueName
13 AnalyticsQueueUrl ← analyticsQueue.queueUrl
14                                     ▷ Research properties
15 HealthResearchQueueId ← researchQueue.id
16 HealthResearchQueueName ← researchQueue.queueName
17 HealthResearchQueueUrl ← researchQueue.queueUrl
18 return whanauRecordQueue, analyticsQueue, researchQueue

```

4.3.1.6 Storage Stack

In this section, we introduce the storage criterion that will be implemented in SST, as described in procedure 6. The inclusion of file upload capability in web applications is a frequently encountered need, and the Serverless Stack (SST) framework streamlines this task by utilizing Amazon S3, a reliable and scalable object storage service. By integrating S3 with SST, developers can effectively manage file uploads without needing to be concerned about the underlying infrastructure. This process consists of multiple crucial stages, all of which contribute to a smooth and safe user experience. The initial stage entails establishing an S3 bucket that acts as the storage location for the uploaded files. This is accomplished by utilizing the Bucket construct in SST, which enables the definition and configuration of the bucket within the application's infrastructure code. To enable seamless communication between the user's front-end and the S3 bucket, the SST framework offers mechanisms for binding the frontend application to the bucket. This connection guarantees that the front-end possesses the required

permissions and abilities to communicate with the bucket, following the principle of least privilege. To ensure secure file uploads, SST utilizes the AWS SDK to create presigned URLs. These URLs offer a secure and temporary method for the front-end to directly upload files to the S3 bucket without revealing sensitive AWS credentials. The server generates a presigned URL that contains permissions and an expiration time. This URL allows the user's browser to securely and efficiently upload the file directly to the S3 bucket. By offloading the data transfer workload to the client-side, the server's load and bandwidth usage are reduced. During this process, it is possible to define the metadata of the file, such as its content type and disposition. This allows for additional customization and control over the uploaded files. The SST framework provides the capability to create user interfaces for file uploads. Developers can incorporate a form into their front-end application, enabling users to choose and upload files. When the form is submitted, it interacts with the presigned URL to securely transfer the file to the S3 bucket. In summary, the SST framework, together with Amazon S3, offers a reliable and user-friendly solution for integrating file upload functionality into serverless applications. This approach not only simplifies the development process but also guarantees secure and efficient handling of file uploads, in line with recommended practices for cloud-based storage and data transfer.

Procedure 6: Storage stack

Definition:

Assume that the stacks listed in the requirements are subsequent stacks already established and are imported. Each stack in the requirements are AWS resources. All requirements are imports of the SST library.

Require: Bucket, StackContext

Result : hAppAgentBucket, hAppServiceBucket

```

1                                     ▷ Build appropriate queues within SQS service
2 hAppAgentBucket ← Bucket(stack, hApp-agent-storage)
3 hAppAgentBucket ← Bucket(stack, hApp-service-storage)
4                                     ▷ Ensure the stack returns an output that is useful to the developer log
5                                     ▷ Storage properties
6 HolochainAppAgentBucket ← hAppAgentBucket.bucketName
7 HolochainAppServiceBucket ← hAppServiceBucket.bucketName
8 return hAppAgentBucket, hAppServiceBucket
```

4.3.1.7 WebSocketApi Stack

Here we define the socket API to be created within SST as seen in the procedure 7. In the Serverless Stack (SST) framework, the WebSocketApi construct offers a simplified and effective method for building WebSocket APIs. These APIs are essential for facilitating real-time, two-way communication between servers and clients in contemporary web applications. By handling the intricacies of WebSocket API configuration, this construct enables developers to concentrate on the fundamental functionality of their applications. The WebSocketApi construct permits the explicit specification of

routes, which serve as the starting points for managing various WebSocket messages. These routes encompass common ones like `$connect`, `$disconnect`, and `$default`, as well as personalized routes for specific message categories. Each route can be linked to an AWS Lambda function, which supplies the computational logic for processing incoming WebSocket messages. The integration takes advantage of AWS Lambda's serverless compute capabilities, guaranteeing scalability and cost efficiency for the WebSocket API. The feature offers support for different authorization mechanisms such as IAM and Lambda authorizers. This guarantees that only clients who have been authenticated and authorized are able to create WebSocket connections and communicate with the API. Developers have the option to assign their WebSocket APIs with custom domains, which improves the branding and user-friendliness of the API. The feature supports domains that are managed by AWS Route 53, and also allows for the importing of existing domains and certificates. The `WebSocketApi` construct offers developers the ability to configure access logs, enabling them to monitor and analyze the traffic and interactions with the WebSocket API. This is essential for ensuring the security and performance of the API. Developers can choose to assign AWS permissions to the entire API or specific routes, ensuring that the associated Lambda functions have the required permissions to interact with other AWS resources in a secure manner. To sum up, the SST framework's `WebSocketApi` construct simplifies the process of creating and managing WebSocket APIs. It offers a variety of functionalities that empower developers to create dependable and adaptable applications. By handling the complex infrastructure components, the framework enables developers to concentrate on delivering valuable functionality within their application logic. This approach is in line with the suggested practices of cloud-native development.

4.4 Evaluation

We have effectively constructed a contemporary infrastructure with the necessary goals in mind to duplicate a centralized system in the cloud. As a result, we have a working version ¹ that allows us to replicate the results ² on Amazon Web Services. The combination of the Serverless Stack (SST) framework and a software methodology like Design Science Research (DSR) or action-based methodology offered an intriguing approach for conducting research in platform development. This section examines the synergy between these elements and explains why their integration is a strong choice for experimental research.

1. **Alignment with Real-World Scenarios:** The constructs of the SST framework, such as `WebSocketApi`, `Api`, and `Auth`, among others, offered abstractions that closely resembled the components found in contemporary, cloud-native applications. This similarity guarantees that the experiments performed using SST are rooted in practicality, showcasing the intricacies and difficulties inherent in the development of real-world platforms.

2. **Facilitation of Iterative Development:** DSR places great importance on the iterative creation of artifacts, constantly evaluating and improving them. The modular design of SST, with its clearly

¹<https://github.com/onahp/paradigm-app>

²The project is divided into sub-directories according to the research phases of this paper

Procedure 7: Websockets stack

Definition:

Assume that the stacks listed in the requirements are subsequent stacks already established and are imported. Each stack in the requirements are AWS resources. All requirements are imports of the SST library.

Require: use, WebSocketApi, DatabaseStack, SNSStack, StackContext

```

1                                     ▷ Define internal service connections to link to web socket
2 hAppAgentConnections ← use(DatabaseStack)
3 topic ← use(SNSStack)
4                                     ▷ Ensure the websocket is linked to the Holochain Agent Service connector
5 api ← WebSocketApi(stack)
6 for each default property in websocket api to i do
7   | timeout ← "20 seconds"
8   | logRetention ← six_months
9   | bind ← [topic, hAppAgentConnections]
10 end
11                                     ▷ Define web socket routes
12 default ← hApp_websocket[default_handler]
13 connect ← hApp_websocket[connect_handler]
14 disconnect ← hApp_websocket[disconnect_handler]
15 sendmessage ← hApp_websocket[sendmessage_handler]
16                                     ▷ Ensure the stack returns an output that is useful to the developer log
17                                     ▷ WebSocket properties
18 WebSocketApiId ← api.id
19 WebSocketApiEndpoint ← api.url
20 WebSocketApiAvailableRoutes ← api.routes

```

defined constructs, fits perfectly with this iterative approach. Researchers can gradually develop and enhance their experimental setups, using SST's constructs to quickly create and test different scenarios. In doing so, they are following the principles of DSR.

3. Enhanced Experimentation Flexibility: Action-based methodologies emphasize the acquisition of knowledge through practical application, typically through interventions and evaluations conducted in real-life situations. The constructs provided by the SST framework offer the necessary adaptability for implementing such interventions. The WebSocketApi construct enables the emulation of live, two-way communication, which is often needed in modern platform development. This versatility guarantees that researchers can modify their experimental configurations to suit various situations, promoting strong and thorough research results.

4. Scalability and Efficient Resource Utilization: When conducting experiments that mimic real-

world situations, a substantial amount of computational resources are often necessary, particularly when scaling up to simulate large-scale applications. SST's serverless paradigm eliminates the need for researchers to concern themselves with the underlying resources, as infrastructure management is abstracted away. This enables researchers to focus on the design of their experiments without being burdened by resource-related concerns. This not only results in cost effectiveness, but also guarantees that the experiments can expand according to the research needs, without the usual burden of managing infrastructure.

5. **Reproducibility and Rigor:** The reproducibility and rigor of experiments are essential in academic research. To achieve this, the SST framework offers a consistent and standardized environment for conducting experiments. With its well-documented constructs and integration with the AWS cloud infrastructure, the framework provides researchers with a reliable means of reproducing and validating experiments. This standardization is crucial as it ensures that findings are rigorously tested and can contribute to the body of knowledge.

In summary, combining the SST framework with methodologies such as DSR or action-based approaches offers a strong approach to conducting research on platform development. This combination not only matches the hands-on and iterative nature of this type of research, but also offers the necessary adaptability, scalability, and rigor to generate valuable and significant research results. Therefore, it serves as an outstanding option for researchers seeking to enhance the field of software engineering by conducting experiments that effectively capture the intricacies of actual platform development.

```

1      $ tree -L 7 -I node_modules
2
3      |-- cdk.context.json
4      |-- package.json
5      |-- packages
6      |   |-- core
7      |   |   |-- package.json
8      |   |   |-- sst-env.d.ts
9      |   |   |-- tsconfig.json
10     |   |-- functions
11     |       |-- handlers
12     |       |   |-- hApp
13     |       |   |   |-- http
14     |       |   |   |-- sample.ts
15     |       |   |   |-- websocket
16     |       |   |       |-- connect.ts
17     |       |   |       |-- default.ts
18     |       |   |-- services
19     |       |       |-- authentication
20     |       |       |   |-- authorizer.ts
21     |       |       |   |-- default.ts
22     |       |       |-- queue
23     |       |       |   |-- consumer
24     |       |       |   |   |-- analytics.ts
25     |       |       |   |   |-- producer
26     |       |       |   |       |-- holochain.ts
27     |       |       |-- topic
28     |       |           |-- publisher
29     |       |           |   |-- hAppAgent.ts
30     |       |           |-- subscriber
31     |       |           |-- hAppIntegratedServiceQueues.ts
32     |       |-- package.json
33     |       |-- sst-env.d.ts
34     |       |-- tsconfig.json
35     |-- pnpm-lock.yaml
36     |-- pnpm-workspace.yaml
37     |-- sst.config.ts
38     |-- stacks
39     |   |-- ApiStack.ts
40     |   |-- AuthStack.ts
41     |   |-- ...
42     |-- tsconfig.json
43
44     17 directories, 32 files

```

Figure 4.1: Research phase 1 : Central infrastructure directory tree

```
1 $ cat package.json
2 {
3     "name": "central-infrastructure",
4     "version": "0.0.0",
5     "private": true,
6     "type": "module",
7     "scripts": {
8         "dev": "sst dev",
9         "build": "sst build",
10        "deploy": "sst deploy",
11        "remove": "sst remove",
12        "console": "sst console",
13        "typecheck": "tsc --noEmit"
14    },
15    "devDependencies": {
16        "@tsconfig/node16": "^16.1.1",
17        "aws-cdk-lib": "2.91.0",
18        "constructs": "10.2.69",
19        "sst": "^2.36.1",
20        "typescript": "^5.2.2"
21    },
22    "workspaces": [
23        "packages/*"
24    ]
25 }
```

Figure 4.2: Research phase 1 : Central infrastructure package

Chapter 5

Research Phase 2 V1: A First Holochain Application

During the initial stage of the development process, our primary goal was to fully understand the core principles of holochain and familiarize ourselves with the development environment. We delved into the structure of holochain and analyzed its unique approach to developing decentralized applications. This phase posed challenges, as acknowledged by both community members and the organization, due to the paradigm shift required to construct systems based on biological principles.

Fortunately, the holochain organization has taken the initiative to clarify and expand on the fundamental concepts [124], furnishing their manual on how to generate a first holochain *"hello world"* application. This serves as a preliminary introduction to the holochain ecosystem. However, during the preliminary development phase, it became evident that the documentation provided was inadequate for novices. Currently, the holochain documentation has been revised to enhance its goal of instructing developers and the community about the underlying architecture of holochain. In addition, it now includes supplementary illustrations of the construction of a sample application with more advanced functionalities compared to the initial *"hello world"* application [124].

5.1 Goals and Objectives

As stated, the goal and objective of this phase was to familiarize ourselves with the underlying concepts of holochain and the technical pillars that build holochain. Here, we introduce and summarize the process and experience of building an initial application within holochain. The following points are our goals for what we want to achieve in this phase.

- Comprehend the fundamental principles of holochain
- Develop a basic *"hello world"* application

- Summarize our findings and challenges during the phase.

By achieving the above goals, we would have sufficient information to dive into building a version two of the application that best incorporates indigenous data sovereignty principles.

5.2 Holochain Formalism

Given that holochain draws inspiration from biological systems [139], the architecture design aligned with the cognitive processes commonly employed in this domain. To ensure coherence and comprehension of the underlying rationale for choosing this architecture, it was necessary to approach the subject from an interdisciplinary perspective that encompassed client/server systems and other systems thinking paradigms. A preliminary draft [145] of the holochain formalism outlines several properties, some of which are emphasized below for their significance in relation to this thesis.¹

1. The **source chain** X_n of a node n within a holochain system Ω_{hc} can be described as the following:

$$X_n \in \Omega_{hc} \quad (5.1)$$

2. M as mentioned in A.3 represents a machine that can run executables, often identified as the set F_{app} :

$$M \equiv F_{app} = \{app_1, app_2, \dots, app_n\} \quad (5.2)$$

3. Given the source chain X_n of n operates in M , the initial entries of X_n in n , comprise various elements of the set $DNA\{e_x, \dots, f_x, \dots, p_x, \dots\}$, where e_x represents the type of entry, f_x denotes the application functions, and p_x reflects the different properties of a system within M :

$$DNA = \{e_1, e_2, \dots, f_1, f_2, \dots, p_1, p_2, \dots, n_x\} \quad (5.3)$$

4. Considering that we start with an initial entry X_n , we also have t_n , referenced in A.3, which corresponds to the second entry of a source chain X_n . Here, t_n represents the set $\{p, i\}$, where p is the public key and i refers to the identification details relevant to the holochain system Ω_{hc} . It should be emphasized that while the types of entries may be identical between various nodes n , their contents differ and are defined as the **agent identity** entry [145]:

$$t_n = \{p, i\} : t_n \in X_n \quad (5.4)$$

¹The formalisms outlined in the preliminary holochain paper are from 2018, are not final and may be modified. However, they offer a glimpse into the planned features and structure of holochain.

5. Since we established $app_n \in F_{app}$ and $\forall e_x \in DNA$, this can be described as the set F_v or the validation functions of the application since $\forall e_x$ are definitions of entry types [145]. This validation logic adheres to the application specifications, which may become more complex as we will discuss later.

$$F_v \implies app_n \in F_{app} \wedge \forall e_x \in DNA \quad (5.5)$$

6. With the validation functions of an agent n , we use the validation functions of the system entry to ensure specificity. $V_{sys}(e_x, e, v)$ where e is the form required by the type of entry definition $e_x \in DNA$. This allows participants in Ω_{hc} of a particular app_n to validate data between peers for malicious behavior.

$$V_{sys}(e_x, e, v) \quad (5.6)$$

Thus, the preceding statement can be characterized as a function where holochain models $V(e, v)$ as a generalized model of a validation function in blockchain where e is represented as a block, v the output of a *proof-of-work* algorithm and V the validity of the output v and the structural integrity of the block e based on the double spending problem [127].

$$V(e, v) \equiv V_x F_v(e_x)(v) \wedge V_{sys}(e_x, e, v) \quad (5.7)$$

7. Here, F_I is introduced as a subset of F_{app} to be distinct from F_v as described in the previous statements, such that $\forall f_x(t)$ there exists a t to $I(t)$ that triggers $f_x(t)$ where $I(t)$ is indicative of a stimulus function that consists of a set of actions available to a user and $f_x(t) \in F_I$.

$$F_I \in F_{app} : \forall f_x(t) \in F_I \exists t := I(t) \quad (5.8)$$

The function F_I is considered to be an exposed function. The functions that are in the set F_{app} and do not belong to F_I or F_v are internal functions which we will define as F_p .

$$\forall F_I \wedge \forall F_v \notin F_{app} \implies F_p \quad (5.9)$$

8. The *DHT* on an authenticated channel C adopts the following properties.

- (a) Δ refers to the DHT state and δ_n is a set $\delta\{key, value\}$ where δ_{key} has the same value as δ_{value} .

$$\Delta\{\delta_1, \delta_2, \delta_3, \dots, \delta_n\} \quad (5.10)$$

(b) F_{DHT} is the set of functions that operate on the DHT

$$F_{DHT} \implies \{dht_{put}, dht_{get}, \dots, dht_f\} \quad (5.11)$$

(c) The $dht_{put}(\delta_{key,value})$ operation adds $\delta_{key,value}$ to the DHT state Δ .

(d) The $dht_{get}(\delta_{key})$ is of equal value to the $\delta_{key,value}$ in the DHT state Δ .

(e) When given two nodes $x, y \in N$, it is assumed that a key value δ_i that exists in the DHT state of node x does not exist in the DHT state of node y . When node y makes a call $dht_{get}(key)$, the key value δ_i is retrieved from the DHT state of x over the authenticated channel X and added to the DHT state y .

$$x, y \in N \wedge \delta_i \in \Delta_x : \delta_i \notin \Delta_y \quad (5.12)$$

9. The DHT_{hc} adopts the following properties.

(a) All pairs of key values $\delta(k, v)$ can only be added, not removed.

$$\forall \delta(k, v) \in \Delta \quad (5.13)$$

(b) The nodes in the neighborhood, denoted n_n , are characterized by the following expression, where $d(x, y)$ represents a unidirectional symmetric metric within the hash space defined by H [145]. The following properties are described as of equal value $H(\delta_{key}) \equiv H(\delta_{value}) \implies A_n = H(pk_n)$ [1], [145]

$$\begin{aligned} V(A_n, q) &= n_n \\ n_n &= q \in \{n_1, n_2, n_3, \dots, n_x\} \\ n_n &= \forall n_i \in n_n \\ n_0 \notin n_n &: d(A_n, A_{n0}) > d(A_n, A_{ni}) \wedge |n_n| = q \end{aligned} \quad (5.14)$$

(c) Each node n surrounded by n_n of size q will **gossip** within its parameterized space to validate the DHT state Δ_n of Ω_{hc} with the original source. All nodes within the DHT are able to discard every $\delta_n \in \Delta$ if the number relative to $d(k, v) > r$ [145].

$$p_x = |\{n_i | d(A_n, \delta_x, k) < d(A_n, \delta_x, k)\}| \quad (5.15)$$

This leads to alterations in the network topology within the DHT, resulting in r divisions of N known as shards, and each shard maintains a replica of the DHT state Δ [145].

10. Given that $\forall n \in N$ assume the DHT state Δ is a subset of D which is the non hash-chain state data previously mentioned. F_{DHT} is accessible to n however they cannot be accessed directly by F_{app} defined in the DNA [145].
11. Assume F_{sys} is the set of functions $\{sys_{commit}, sys_{get}, sys_{send}, \dots, sys_f\}$. The properties are incomplete but contain at the very least the following:
- (a) The $sys_{commit}(e)$ function employs the validate functions $V(e, v)$ to add e to X and if successful calls the following operational function $dht_{put}(H(e), e)$ to add to the DHT state Δ . $sys_{send}(A_{to}, m)$ initiates the $apprecieve(A_{from}, m)$ among nodes n with m being the message. This is referred to as **node-to-node messaging** [145].

$$sys_f = \begin{cases} sys_{commit}(e) = dht_{put}(H(e), e) & \text{if } V(e, v) \\ sys_{get}(k) = dht_{get}(k) & \text{if } x, y \in N \wedge \delta_x \in \Delta_x : \delta_y \notin \Delta_y \\ sys_{send}(A_{to}, m) = apprecieve(A_{from}, m) & \text{if } m \in n : n \in \Omega_{hc} \end{cases} \quad (5.16)$$

The functions within F_{app} specified in the DNA are capable of invoking functions in F_{sys} [145].

12. Entries σ that are added to the DNA can be of type **private** and are enforced to participate outside of Δ [145].

$$\sigma_{private} \iff \sigma_i \notin \Delta \quad (5.17)$$

The process of inserting an entry into the DHT requires finding an appropriate node for that entry. According to the authors [145], the time complexity of this task is an approximate assumed $c + [\log m]$, derived from [1]. Consequently, the nodes responsible for validating the entry ², is assumed $v(n \cdot m)$ and represents the specific validation logic of an application ³ according to their own requirements, with q representing the neighboring nodes and r denoting the number of copies or r nodes [145]. This assumes that the average complexity of the holochain system Ω_{hc} can be defined as:

$$\begin{aligned} DHT_{lookup} &\approx c + [\log(m)] \\ Entry_{validation} &\approx v(n \cdot m) \\ \Omega_{hc} &\approx n \cdot (\log(m) + v(n \cdot m)) \end{aligned} \quad (5.18)$$

²Application specific logic relative to the holochain system

³The validation logic is unknown as complexity is determined by the developer of the specific holochain app

The time complexity varies according to the specific application being developed, but it is assumed that the probability of participation of nodes is influenced by the assumption that the hash function evenly distributes the hash uniformly [145]. While there is still a lot to explore about the formalism of holochain, this provides the foundation for developing a holochain application with the core principles in consideration.

5.3 Defining Architecture

With a better understanding of the fundamental principles of the holochain system, we were able to quickly move forward with designing a basic holochain application. This allowed us to dive deeper into the system's inner workings. It was crucial for us to understand these principles to ensure our application's effectiveness. We adopted a high-level perspective of our application's expectations and compared it with the official documentation to translate these principles into code for our understanding. This approach was beneficial, as it helped us maintain a clear view of the overarching concepts while simultaneously developing the detailed components of the holochain application.

Our first application based on the design of the holochain should have a general process as described 5.1.

Following the structure depicted in Figure 5.1, we began developing small components of the holochain application focused on unit test results which encouraged fine-grained techniques to improve efficiency, as well as developer experience and code quality [107], [122].⁴. Initially, our goal was to construct the entire application using black-box methods to gain practical experience and build conceptual relationships from theory. However, this approach proved to be too time consuming. Therefore, we decided to focus on meeting the minimum specifications required for a minimum viable product using unit test practices 5.2, which demonstrate that the components we sought to build within the holochain are functional at the very least.

In addition, our goal was to outline straightforward scenarios to guide us in achieving the desired results through an emphasis on unit tests, and we established our process 5.3 using a straightforward structural flow to facilitate replication and repeatability. This approach advanced our application development and enhanced our comprehension of the application's practical elements beyond its theoretical aspects.

5.4 Building a Holochain Application

We initiated the development of our application focusing on unit tests as mentioned above. This method proved to be convenient for us because of node's straightforward implementation for executing test cases within a project initiated by the scaffolding tool, along with holochain's support for such features. Our objective was to quickly outline the desired output without creating any UI components.

⁴Test driven development generally has pro's and con's depending on the type of application being built.

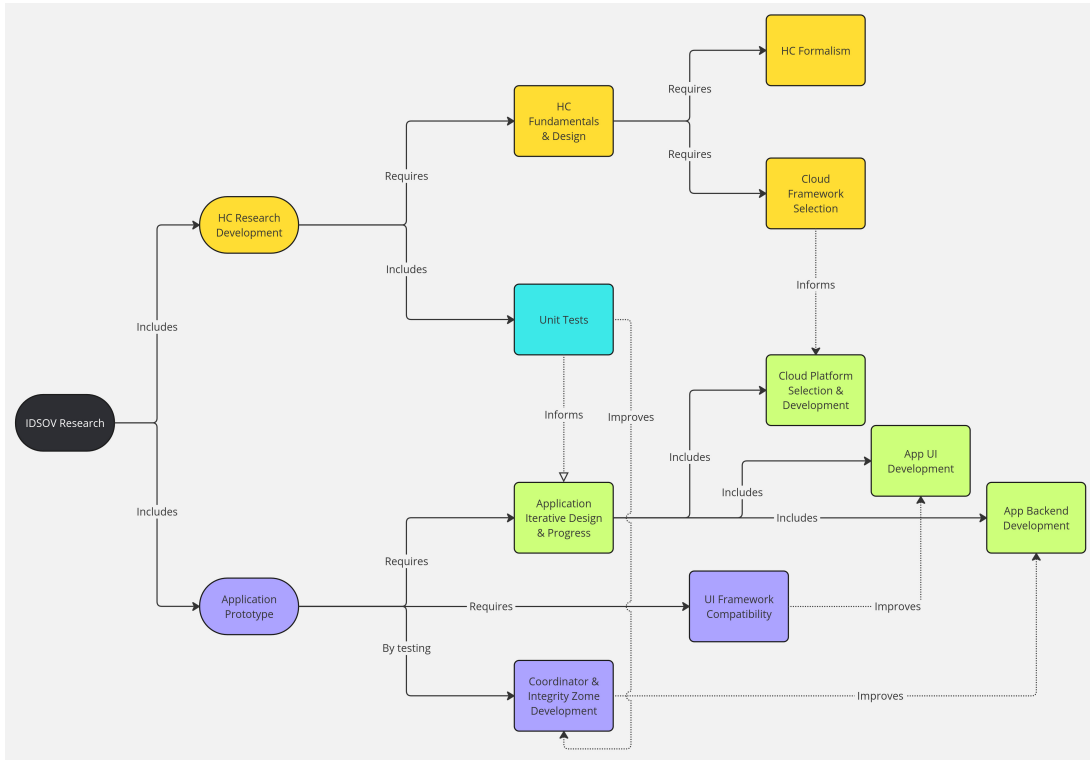


Figure 5.2: Research application overview development

Based on our tests, it was essential to complete the remaining operations to update the entries as shown 11. Despite the absence of a visual UI component, we deduced that the operation was successful based on the changes observed, the output transferred between participants, and the validation rules affirming the entries' validity according to the established integrity guidelines.

Finally, we display the tests 12 for the delete operation to complete the set of *CRUD*-like functionalities for managing entries. It is crucial to mention that as we got acquainted with these operations, we realized that the validation rules we established defined the application's health, integrity, and limits. The flexibility of these rules, controlled by the developer and managed by the conductor, can indeed grow in complexity. However, we discovered that basic operations covered most of our requirements for general application purposes for most of our needs, similarly to centralized systems.

Our motivation for the previously mentioned tests arose from the need to create both integrity and coordinator zones in accordance with the guidelines set by the holochain organization. The zone functions, along with entries, hashes, and other elements, are crucial components of a holochain application. Hence, we designed these components fundamentally without focusing on specific data outputs. Our main objective was to preserve the types of data integrity definitions, irrespective of their

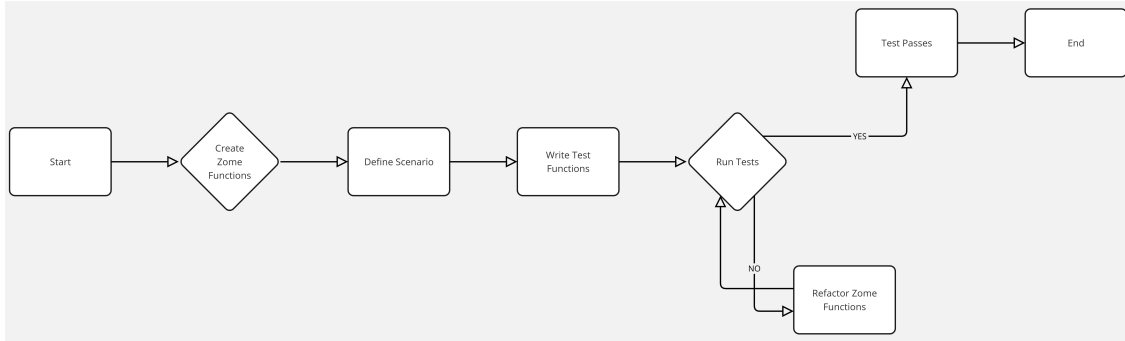


Figure 5.3: Software development process flow

Procedure 8: General DNA integrity and coordinator scenario**Definition:**

Assume $\text{Agent}[x_1, \dots, x_n]$ represents the total number of agents involved in a holochain network, with $\text{Agent}(x)$ denoting a single agent,

A_s the *appSource*,

T_p the *testAppPath*,

H_p the *pathToProject.happ*,

R_x the created record by $\text{Agent}(x)$,

S_a the *scenario.addPlayersWithApps*,

S_s the *scenario.shareAllAgents*.

Require: vitest, @holochain/tryorama, @msgpack/msgpack

Data: D^l where D^l represents the test data that has been either generated or imported into the repository.

Zone: $Z^i \wedge Z_c$ where Z^i is the integrity zome and Z_c is the coordinator zome under examination.

```

1 if  $D^l > 0$  and  $Z^i \wedge Z_c$  is an existing callable cell then
2   while test(case) do
3      $T_p \leftarrow H_p$ ;
4      $A_s \leftarrow \{appBundleSource : T_p\}$ ;
5      $\text{Agent}[x_1, x_2, x_3, \dots, x_n] \leftarrow S_a([A_s \cdot \text{Agent}[x_n]])$ ;
6     await  $\rightarrow S_s$ ;
7      $R_x \leftarrow \text{importedZomeFunction}(\text{Agent}(n).cells[i])$ ;
8     assert.ok( $R_x$ );
9   end
10 end
  
```

Procedure 9: Zome functions and entries scenario - create

Definition:

Assume $\text{Agent}[x_1, \dots, x_n]$ represents the total number of agents involved in a holochain network, with $\text{Agent}(x)$ denoting a single agent,

A_s the *appSource*,

T_p the *testAppPath*,

H_p the *pathToProject.happ*,

R_x the created record by $\text{Agent}(x)$,

S_a the *scenario.addPlayersWithApps*,

S_s the *scenario.shareAllAgents*.

Require: vitest, @holochain/tryorama, @msgpack/msgpack

Data: D^l where D^l represents the test data that has been either generated or imported into the repository.

Zome: $Z^i \wedge Z_c$ where Z^i is the integrity zome and Z_c is the coordinator zome under examination.

```

1                                     ▷ Create an entry
2 if  $D^l > 0$  and  $Z^i \wedge Z_c$  is an existing callable cell then
3   while test (create health record) do
4      $T_p \leftarrow H_p$ ;
5      $A_s \leftarrow \{appBundleSource : T_p\}$ ;
6      $\text{Agent}[a, b] \leftarrow S_a([A_s, A_s])$ ;
7     await  $\rightarrow S_s$ ;
8      $R_a \leftarrow \text{createHealthRecord}(\text{Agent}(a).cells[0])$ ;
9      $R_b \leftarrow \text{createHealthRecord}(\text{Agent}(b).cells[0])$ ;
10    assert.ok( $R_a$ );
11    assert.ok( $R_b$ );
12  end
13 end

```

form. Therefore, our emphasis was on replicating the definition type rather than its various contents. At the time of writing, an HC scaffolding tool ⁵ was implemented to enhance the developer experience and minimize build-time errors, enabling the developer to concentrate on building the application itself. By using the developer scaffolding tool along with our recent test development experience with zome modifications, we gained some understanding of the underlying mechanics of holochain. This inspired us to develop our own application with the goal of deepening our comprehension of the developer experience associated with creating a holochain application.

Following the instructions given promptly, we were able to quickly generate a basic "hello world"

⁵<https://github.com/holochain/scaffolding>

Procedure 10: Zome functions and entries scenario - create and read**Definition:**

Assume $\text{Agent}[x_1, \dots, x_n]$ represents the total number of agents involved in a holochain network, with $\text{Agent}(x)$ denoting a single agent,

A_s the *appSource*,

T_p the *testAppPath*,

H_p the *pathToProject.happ*,

R_x the created record by $\text{Agent}(x)$,

S_a the *scenario.addPlayersWithApps*,

S_s the *scenario.shareAllAgents*.

Require: vitest, @holochain/tryorama, @msgpack/msgpack

Data: D^l where D^l represents the test data that has been either generated or imported into the repository.

Zome: $Z^i \wedge Z_c$ where Z^i is the integrity zome and Z_c is the coordinator zome under examination.

```

1                                     ▷ Create and read entry
2 if  $D^l > 0$  and  $Z^i \wedge Z_c$  is an existing callable cell then
3   while test (create health record) do
4      $T_p \leftarrow H_p$ ;
5      $A_s \leftarrow \{appBundleSource : T_p\}$ ;
6      $\text{Agent}[a, b] \leftarrow S_a([A_s, A_s])$ ;
7     await  $\rightarrow S_s$ ;
8     sample  $\leftarrow \text{sampleHealthRecord}(\text{Agent}(a).cells[0])$ ;
9      $R_a \leftarrow \text{createHealthRecord}(\text{Agent}(a).cells[0], \text{sample})$ ;
10    await  $\rightarrow \text{dhtSync}(\text{Agent}[a, b], \text{Agent}(a).cells[0].cell_id[0])$ ;
11    record  $\leftarrow \text{Agent}(b).cells[0].callZome(\{$ 
12      zome_name  $\leftarrow dna\_zome,$ 
13      fn_name  $\leftarrow \text{update\_health\_record},$ 
14      payload  $\leftarrow \text{updateEntryContent}$ 
15     $\});$ 
16    assert.deepEqual(sample, decode(record.entry));
17  end
18 end

```

application in holochain 5.5. This enabled us to quickly build an application with two participating agents. We were able to examine the source chain, entries, and connections between each node, observing their interactions. It was evident that the installation due to rust's compiler took some time to install given the nix environment installations and WASM.

```
1     $ nix run github:holochain/holochain#hc-scaffold -- example hello-world
2
3     Setting up nix development environment...
4     Warning: creating lock file
5     'holochain/apps/hello-world/flake.lock'
6     Initialized empty Git repository in
7     holochain/apps/hello-world/.git/
8
9     Example "hello-world" scaffolded!
10
11    Run the example app with:
12
13    cd hello-world
14    nix develop
15    npm install
16    npm start
```

Figure 5.4: Research phase 2 V1 : Holochain scaffolding tool example

Once the application was loaded, we were provided with a developer tool that allowed us to interact with our application in real time 5.6. Using these features, we were able to inspect existing call functions defined by the scaffolding tool and observe the requests made by each entry in the developer tool. This demonstration allowed us to understand how a UI component communicates with the DNA and the latency between the back-end and front-end or the client and conductor. In the presented application sample, it was straightforward for us to visualize some of the components we intended to build, although we did note that the user interface components were not as clearly structured, which we kept in mind for future phases.

Upon inspecting the source chain 5.7 using our developer tool, we noticed that the nodes were organized in the following sequence. The node graph 5.8 displayed a summarized depiction of actions systematically performed on the source chain. The origin chain begins with a cluster containing the DNA component of the application, followed by the ensuing events within the holochain application shortly thereafter. This supported our findings on how the source chain operates within the DHT_{hc} [145]. When examining the nodes illustrated on the graph using our developer tool, we observed the related entries that were generated, as shown in the chain graph of a single cell 5.9. These generated entries showcased the various actions executed by a single agent, connected by its entry type and entry id. Although this application served as a prototype for what we aimed to develop, we noted the display of entries within the DHT and the representation of the source chain. At this stage, we aimed to develop an initial application that showcased an arbitrary result, drawing from our varied experiences

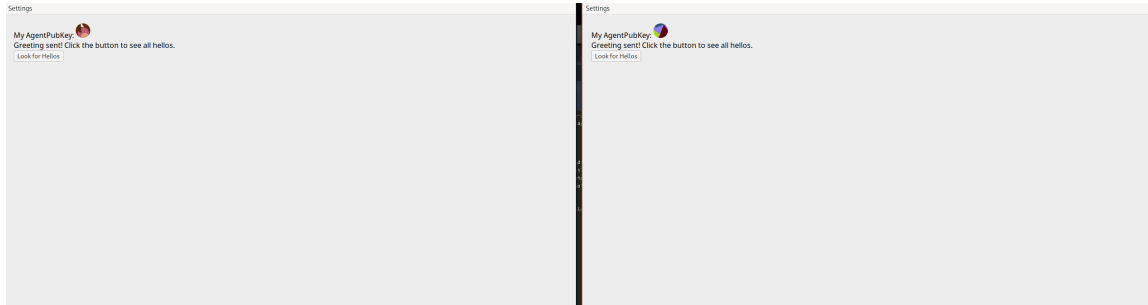


Figure 5.5: Active "hello world" application with two nodes

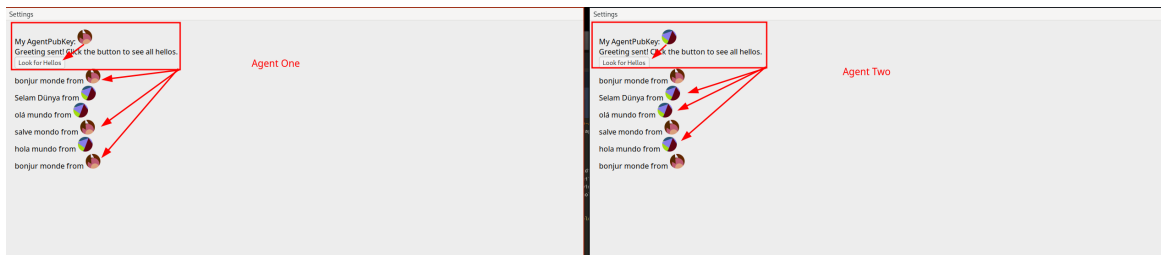


Figure 5.6: Active "hello world" application with multiple entries

during testing and our observations of the pre-built application presented in the organization's sample. This application would serve as our initial step towards developing a more polished application that we demonstrate later on in this paper.

At the start of developing the first version of our application, we used the scaffolding tool to get the initial application up and running, similar to our previous demonstration with the "hello world" sample application provided by the holochain organization. However, we added our own customizations to the types of data that we wanted to include. The data were arbitrary rather than specific. Our objective, as with our tests, was to guide the development of the application through our scenario test cases 5.3. These tests informed us about the types of entries we wanted to create and the kind of application we aimed to develop for the first version. We outline our progress and the steps we took to achieve it. We called our first version of the application *Iwi*.

Using the scaffolding tool mentioned above, we managed to create the initial version of our application, which included a few random integrity categories. The main objective of this effort was to evaluate, throughout our development trials, how our activities were perceived from a user narrative standpoint. Our priority was not on emphasizing UI/UX elements, but rather on achieving a basic viable product that could effectively showcase the process of managing entries visibly. This strategy enabled us to produce valuable improvements on the application's functionality and usability, facilitating iterative enhancements.

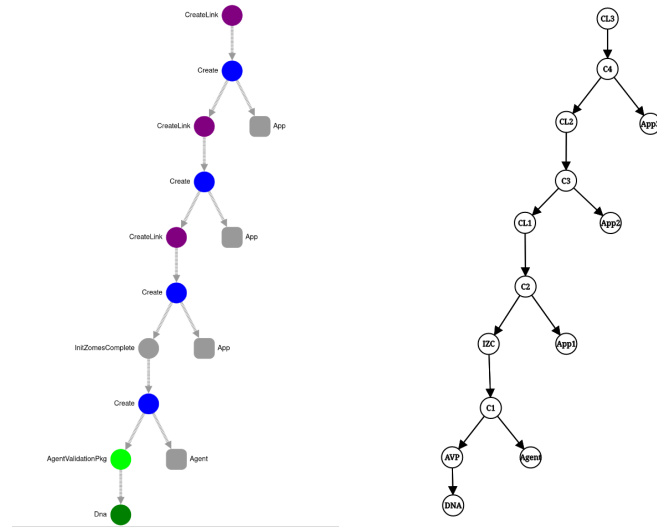


Figure 5.7: Sample "hello world" application Source chain

5.4.1 IDSOV Iwi Integrity Type Definitions

In our effort to create a prototype of a simple indigenous application, we aimed to incorporate a crucial requirement from MASov's principles of language preservation. This requirement ensures that the data are inclusive of language, principles, resources, and rights in their respective terms of its origin [116]. Our initial version of the application code includes indigenous terminology 13. We opted to use indigenous terms for naming the struct types in our application, while the type definitions were provided in English translation. Our reasoning was to maintain simplicity in the application, and we structured the entry types as *String* to permit arbitrary values, facilitating easier testing.

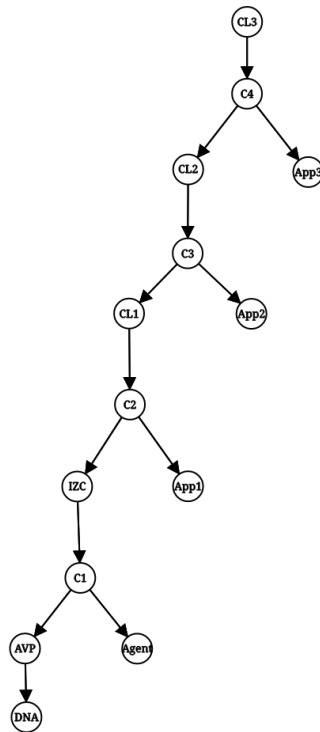


Figure 5.8: Sample "hello world" application node graph

Procedure 13: IDSOV integrity types

Definition:

All definitions mentioned below are translations from māori to english.

Patoi ;

⇒ *Announcement*,

Katoitoi ;

⇒ *Response*,

Panui Katoa ;

⇒ *All Announcements*

Require: hdi

```

1 macros ← [hdk_entry_helper, derive(Clone, PartialEq)]
2 pub struct Panui {
3   title ← type ← String,
4   content ← type ← String,
5 }
6 macros ← [hdk_entry_helper, derive(Clone, PartialEq)]
7 pub struct Katoitoi {
8   katoitoi ← type ← String,
9   panui_hash ← type ← ActionHash,
10 }

```

The *Katoitoi* struct type has a property named *panui.hash* of type *ActionHash*. The *ActionHash* is linked with the *Panui* struct, meaning that each announcement is an action, and every related response to an announcement includes the *ActionHash* of that announcement, thereby connecting the two. This establishes a direct association between each response and its respective announcement, similar to parent-child relationships in relational databases. Unlike the *EntryHash*, which is primarily used to fetch the actual entry content [124], which was not our aim on the two.

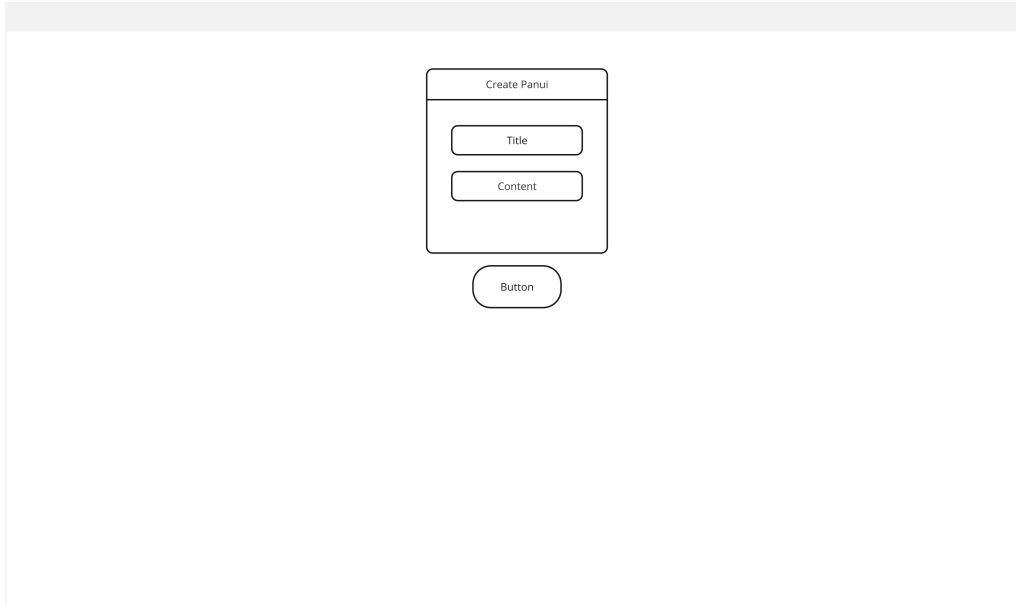


Figure 5.10: IDSOV iwi first design

5.4.2 IDSOV Iwi Simple UI Design

Similarly to many initial application developments, our procedures aimed to create a preliminary application with functional services. While user design is crucial, we did not emphasize it as a developer story. Instead, we opted for a straightforward design that accurately reflected the data transmitted from the back-end to the client. Our front-end development process is presented in its simplest form 5.10 to ensure ease of use and accessibility for all users, regardless of their technical proficiency or familiarity with the application. We recognize that prioritizing functionality over user design can result in a less satisfactory user experience. Ignoring the developer narrative and choosing a simplistic design might cause users, who prefer more attractive and user-friendly interfaces, to lose interest. Streamlining the UI development process to accommodate users of all technical skill levels might miss the opportunity to incorporate more sophisticated features that could improve the overall user experience. Taking into account these differing viewpoints, we understand that achieving a balance between functionality, user

design, and technical innovation is essential for creating a successful application that fulfills the varied needs and expectations of users, which is our goal for the final version of the application. Therefore, based on the design depicted in 5.10, we implemented slight improvements to present the data content as shown here for the sake of progression 5.11.

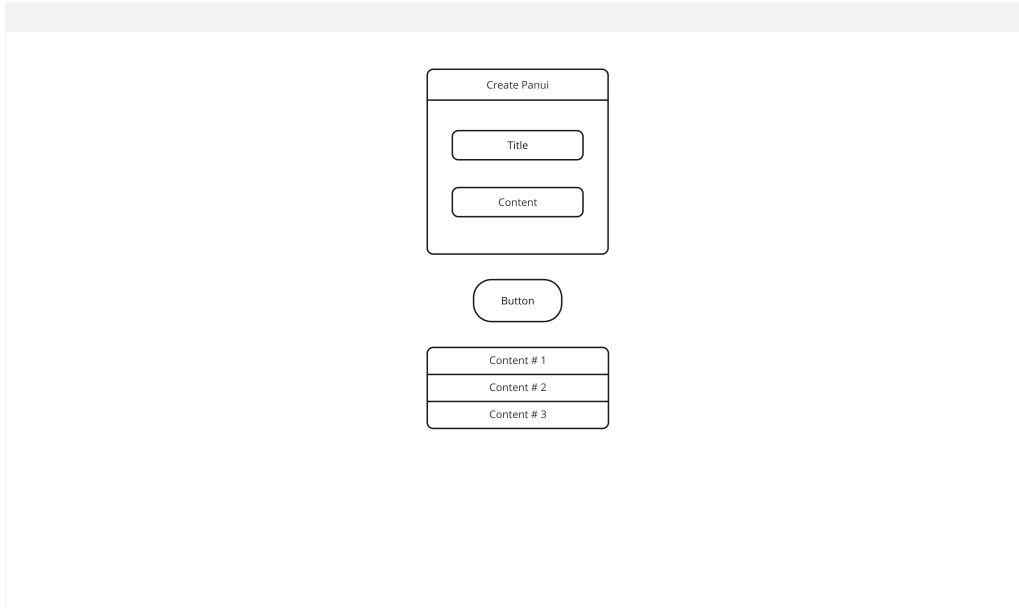


Figure 5.11: IDSOV iwi minor improvements

5.4.3 IDSOV Iwi Build

During the development of the final version⁶ of our application *Iwi* using the holochain organization scaffolding tool, we encountered several issues. The UI elements that we aimed to create were available only in a few frameworks, specifically *Svelte*, *Lit*, and *Vue*. Ultimately, we chose *Svelte* due to its popularity and the simplicity of its evolving features, which we managed to use to produce our final design in display 5.13. Once the backend of the application was successfully configured with the coordinator and integrity zones running, we redirected our efforts towards linking the front-end client with the holochain zone functions. We adopted a unit testing approach to guarantee the system's functionality in scenario's. This strategic choice enabled us to advance effectively, despite facing difficulties with the front-end libraries, which sometimes impeded our progress. Although we faced this challenge, we deliberately decided not to focus on the UI elements for the initial prototype. We deemed it more crucial to build a robust foundation for the application's core functionality. Consequently,

⁶The completed initial version of the Iwi application can be accessed at <https://github.com/onahp/paradigm-app> in the `idsov-iwi` folder of the repository.

we allocated more resources to enhancing the communication protocols between the front-end and holochain zone functions. This effort greatly boosted the system's overall performance and reliability. Our success was evident in the passing of our unit test cases and the minimization of unnecessary data processing delays, enabling us to concentrate on the next stages of development.

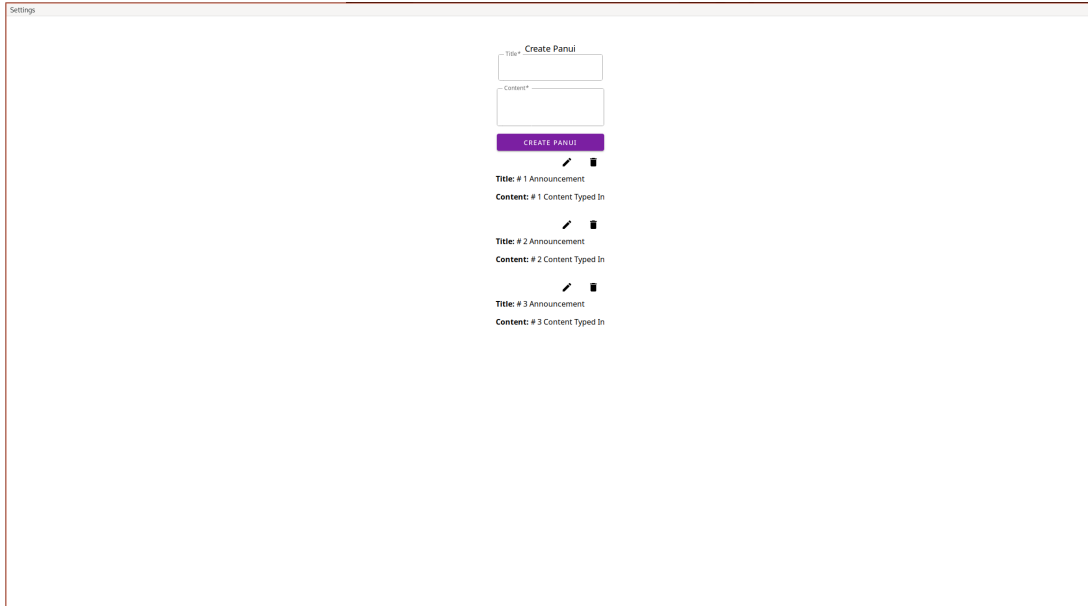


Figure 5.12: IDSOV iwi application display with entries

Some may contend that emphasizing the UI elements in the initial prototype could have enhanced the user experience and provided a clearer visualization of the application's features. We do acknowledge that concentrating exclusively on the backend and communication protocols risks overlooking the user interface which may impede on the final version of the application, which is essential for attracting and retaining users. Additionally, we recognize that while unit testing is an effective approach, it may not always account for real-world challenges and user interactions that can arise during development where other methods might be more reliable like regression testing, test-driven development, and E2E for more complete applications with lower time constraints. Relying solely on passing unit tests might not uncover all potential problems that could impact the system in a production environment. Therefore, we acknowledged that the iwi application could be seen as somewhat lacking in features, with significant potential for enhancement as seen in 5.13 where both agents are able to delete and modify each of the posted records from either of them.

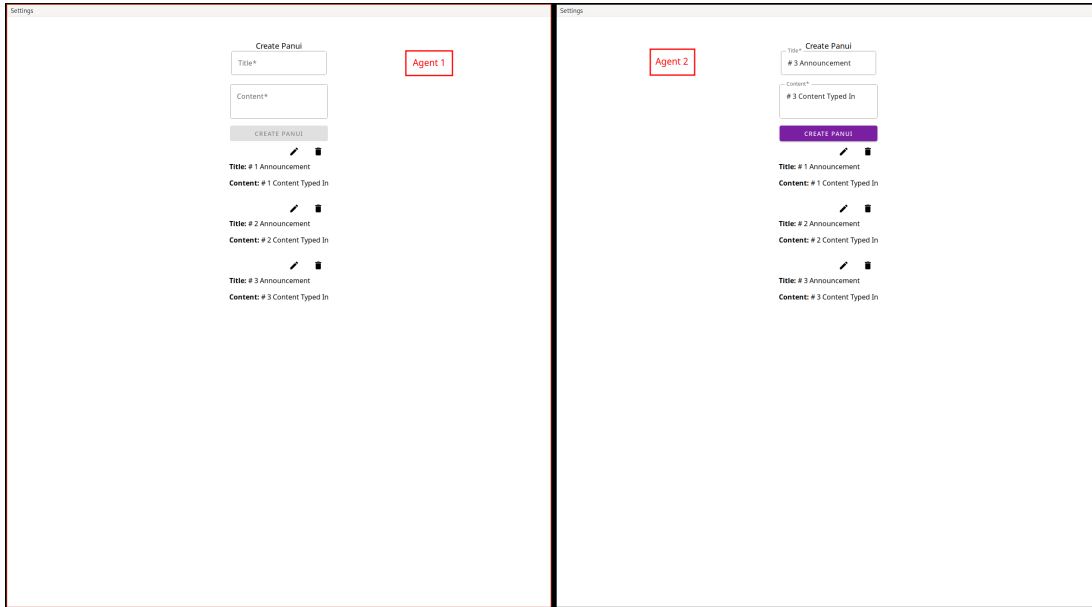


Figure 5.13: IDSOV iwi application display with entries and two agents

5.5 Challenges and Findings

In the process of creating the initial holochain application, we found it to be quite fulfilling. However, we also identified several significant limitations that surfaced during development. These limitations became evident when we considered the prerequisites for building holochain applications, which include expertise in the Rust programming language, knowledge of holochain architecture, familiarity with the supported front-end libraries, and a readiness to adopt a decentralized approach. Grasping these requirements was crucial for us to advance through our development phases. We particularly struggled to understand the core structure of the holochain, as it is unconventional to think of software as decentralized agents. Most modern technology is based on a centralized model, which users are more familiar with. Adapting to this decentralized approach can be challenging. Additionally, the learning curve was steep due to the intertwining of concepts with biological systems. Initially, these concepts were adaptable, but the lack of documentation, tools, and flexibility eventually became obstacles. For instance, the similarities between cells and agents, or the DHT operations used by cells to validate entries, can be confusing. Nonetheless, we view these challenges as opportunities for improvement.

5.6 Evaluation

Developing a holochain application comes with its own set of constraints and difficulties. However, the potential for holochain to evolve into a fully developed platform is present, offering developers an improved experience when creating sophisticated applications. Currently, the organization is working to improve the developer experience. However, this effort is not as mature as the more established technologies in the centralized world. Therefore, the developer experience and growth rate are unlikely to see a significant boost given holochain's lesser popularity relative to other modern technologies. Despite this, at the current moment, holochain holds some potential as a valuable technology within the decentralized ecosystem.

Procedure 11: Zome functions and entries scenario - create and update**Definition:**

Assume $\text{Agent}[x_1, \dots, x_n]$ represents the total number of agents involved in a holochain network, with $\text{Agent}(x)$ denoting a single agent,

A_s the *appSource*,

T_p the *testAppPath*,

H_p the *pathToProject.happ*,

R_x the created record by $\text{Agent}(x)$,

S_a the *scenario.addPlayersWithApps*,

S_s the *scenario.shareAllAgents*.

Require: vitest, @holochain/tryorama, @msgpack/msgpack

Data: D^l where D^l represents the test data that has been either generated or imported into the repository.

Zome: $Z^i \wedge Z_c$ where Z^i is the integrity zome and Z_c is the coordinator zome under examination.

```

1                                                                 ▷ Create entry
2 if  $D^l > 0$  and  $Z^i \wedge Z_c$  is an existing callable cell then
3   while test (create record and update record) do
4      $T_p \leftarrow H_p$ ;
5      $A_s \leftarrow \{ \text{appBundleSource} : T_p \}$ ;
6      $\text{Agent}[a, b] \leftarrow S_a([A_s, A_s])$ ;
7     await  $\rightarrow S_s$ ;
8      $\text{sample} \leftarrow \text{sampleHealthRecord}(\text{Agent}(a).\text{cells}[0])$ ;
9      $R_a \leftarrow \text{createHealthRecord}(\text{Agent}(a).\text{cells}[0], \text{sample})$ ;
10    assert.ok( $R_a$ );
11                                                                 ▷ Update entry
12     $\text{originalActionHash} \leftarrow R_a(\text{signed\_action}.\text{hashed}.\text{hash})$ ;
13     $\text{updateEntryContent} \leftarrow \text{sampleHealthRecord}(\text{Agent}(a).\text{cells}[0])$ ;
14     $R_a(\text{update}) \leftarrow \text{Agent}(a).\text{cells}[0].\text{callZome}(\{$ 
15       $\text{zome\_name} \leftarrow \text{dna\_zome}$ ,
16       $\text{fn\_name} \leftarrow \text{update\_health\_record}$ ,
17       $\text{payload} \leftarrow \text{updateEntryContent}$ 
18     $\});$ 
19    assert.ok( $R_a(\text{update})$ );
20                                                                 ▷ Sync participants and read updated entry
21    await  $\rightarrow \text{dhtSync}(\text{Agent}[a, b], \text{Agent}(a).\text{cells}[0].\text{cell\_id}[0])$ ;
22     $\text{record} \leftarrow \text{Agent}(b).\text{cells}[0].\text{callZome}\{\text{zome}, \text{function}, \text{payload}\}$ ;
23     $\text{updatedRecord} \leftarrow \text{Agent}(b).\text{cells}[0].\text{callZome}(\{$ 
24       $\text{zome\_name} \leftarrow \text{dna\_zome}$ ,
25       $\text{fn\_name} \leftarrow \text{get\_latest\_health\_record}$ ,
26       $\text{payload} \leftarrow R_a(\text{update}).\text{signed\_action}.\text{hashed}.\text{hash}$ 
27     $\});$ 
28    assert.deepEqual( $R_a, \text{decode}(\text{updatedRecord}.\text{entry})$ );
29  end
30 end

```

Procedure 12: Zome functions and entries scenario - create and delete**Definition:**

Assume $\text{Agent}[x_1, \dots, x_n]$ represents the total number of agents involved in a holochain network, with $\text{Agent}(x)$ denoting a single agent,

A_s the *appSource*,

T_p the *testAppPath*,

H_p the *pathToProject.happ*,

R_x the created record by $\text{Agent}(x)$,

S_a the *scenario.addPlayersWithApps*,

S_s the *scenario.shareAllAgents*.

Require: vitest, @holochain/tryorama, @msgpack/msgpack

Data: D^l where D^l represents the test data that has been either generated or imported into the repository.

Zone: $Z^i \wedge Z_c$ where Z^i is the integrity zome and Z_c is the coordinator zome under examination.

```

1                                     ▷ Create entry
2 if  $D^l > 0$  and  $Z^i \wedge Z_c$  is an existing callable cell then
3   while test (create record and update record) do
4      $T_p \leftarrow H_p$ ;
5      $A_s \leftarrow \{appBundleSource : T_p\}$ ;
6      $\text{Agent}[a, b] \leftarrow S_a([A_s, A_s])$ ;
7     await  $\rightarrow S_s$ ;
8     sample  $\leftarrow \text{sampleHealthRecord}(\text{Agent}(a).cells[0])$ ;
9      $R_a \leftarrow \text{createHealthRecord}(\text{Agent}(a).cells[0], \text{sample})$ ;
10    assert.ok( $R_a$ );
11    await  $\rightarrow \text{dhtSync}(\text{Agent}[a, b], \text{Agent}(a).cells[0].cell_id[0])$ ;
12                                     ▷ Delete entry
13    deleteActionHash  $\leftarrow \text{Agent}(a).cells[0].callZome(\{$ 
14      zome_name  $\leftarrow dna\_zome,$ 
15      fn_name  $\leftarrow \text{delete\_health\_record},$ 
16      payload  $\leftarrow R_a.signed\_action.hash$ 
17     $\});$ 
18    assert.ok(deleteActionhash);
19                                     ▷ Sync participants
20    await  $\rightarrow \text{dhtSync}(\text{Agent}[a, b], \text{Agent}(a).cells[0].cell_id[0])$ ;
21  end
22 end

```

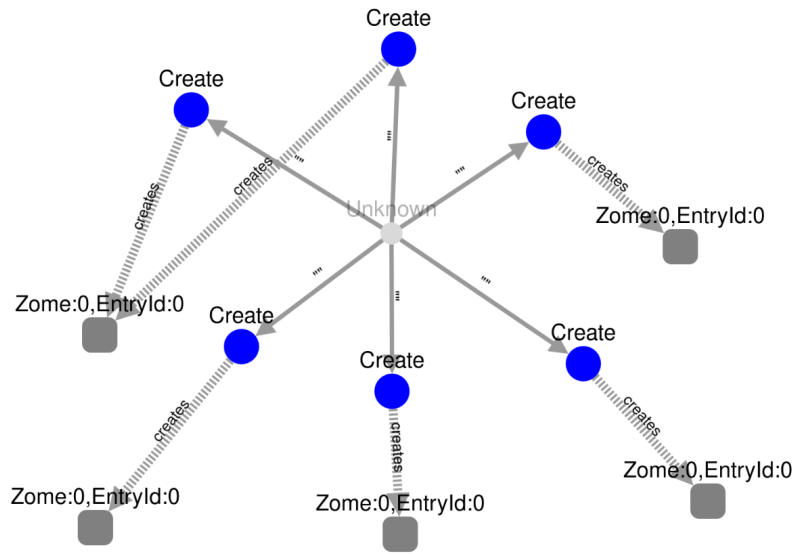


Figure 5.9: Sample "hello world" entries on the entry graph

Chapter 6

Research Phase 2 V2: IDS Holochain Application Prototype

In the second version of phase 2 of our project, our main objective was to integrate the principles of indigenous data sovereignty into our holochain application. Understanding the vital importance of indigenous data sovereignty and its impact on the development of holochain products was crucial for our success. Indigenous data sovereignty represents the inherent right of indigenous communities to manage, access, and control the use and distribution of their data. Although we lacked the relevant stakeholders to make significant progress, we relied on existing literature on indigenous data sovereignty principles, particularly those of the indigenous māori. This engagement was crucial to ensure that our prototype not only respects and upholds their rights and principles but also attempts to address their specific needs and concerns to the best of our ability. Additionally, our development approach is committed to ethical and culturally sensitive data practices, which include incorporating diverse perspectives and knowledge systems from indigenous communities.

By emphasizing the sovereignty of indigenous data in the development of our holochain prototype, we ensure adherence to ethical standards while fostering a more inclusive and equitable technology development process. Building on our previous iteration of the iwi application, we aimed to enhance the existing features, UI components, and integrity definitions. This required us to expand the entry-types from the previous application and update the UI client libraries to reflect what an application influenced by indigenous perspectives would look like. Consequently, we gathered extensive literature and research to document the improvements made from the earlier version to the final version of our application. Our findings and results are presented in this chapter.

6.1 Goals and Objectives

The goal of this phase was to utilize the insights gained from developing the initial version of the holochain application to create a new version that adhered more closely to the principles of indigenous

data sovereignty. Consequently, the application included a design system that aligned well with IDS principles, working alongside MASov to produce a prototype that was culturally acceptable. We recognized that the prototype’s development faced significant constraints due to the fundamental differences between the intended holochain and IDS governance principles, but we aimed to incorporate the available resources in conjunction with existing literature.

- **Language and Modality:** The application should be intuitive to its purpose, allowing individuals to engage effortlessly within its ecosystem.
- **Incorporating Indigenous Data Sovereignty:** The application should integrate IDS principles into its design to more accurately reflect real-world practices aligned with cultural normality.
- **Developing a Minimum Viable Product:** Considering the potential complexity of holochain applications, creating an MVP that includes the previously stated objectives should suffice for this experimental phase. However, there should be notable advancements from the initial version of the application to the subsequent one.

The screenshot shows a 'Settings' page with a 'Create Profile' form. The form includes the following fields and elements:

- Avatar:** A circular icon with a plus sign (+).
- Nickname*:** A text input field with a placeholder and a note 'Min. 3 characters' below it.
- Maunga*:** A text input field.
- Moana*:** A text input field.
- Nō Hea Koe*:** A text input field.
- Whanāu*:** A text input field.
- Ingoa*:** A text input field.
- Create Profile:** A blue button at the bottom of the form.

Figure 6.1: Adapted changes of the profile

6.2 Defining Requirements

The following requirements are the guiding principles that incorporate a synthesized version of the MASov principles ¹. We will endeavor to develop a holochain application that prioritizes indigenous data sovereignty principles. It is worth noting that due to time constraints and the need to learn holochain for building the application, there are recognized limitations. Therefore, this phase will clearly define requirements that do not compromise the experiment while offering enough to demonstrate a functional example of our intended creation.

- The holochain application should incorporate the māori language as part of its indigenous approach to data sovereignty, emphasizing inclusivity.
- The holochain application should have the capability to exchange data among peers or agents within the application's network.
- The holochain application should incorporate a profile sign-up that resembles a Pepeha ², serving as a foundation for promoting cultural awareness and showcasing indigenous values to individuals connected within its network.

These characteristics should meet the DSR standards to show understanding and confirmation of the indigenous viewpoint that is significant to the stakeholder. Although there may not be enough time to further develop and repeat the process cycle, the initial iterative cycle signifies a cycle within the design or action methodology. This allows us to confirm the existence of the cycle and acknowledge its limitations within the scope of this thesis during the methodology process.

MASov Principles	Responsible Algorithm Principles
Rangatiratanga	Fairness, Responsibility, Trust, Dignity
Whakapapa	Transparency, Responsibility, Sustainability
Whanaungatanga	Transparency, Fairness, Trust, Solidarity
Kotahitanga	Fairness and justice, solidarity
Manaakitanga	Responsibility, privacy, trust, dignity
Kaitiakitanga	Transparency, Privacy, Trust, Sustainability, Solidarity

Table 6.1: Adopted alignment of MASov principles with algorithm principles that can translate into requirements for building a holochain application [116]

¹The algorithmic principles in alignment with MASov principles can be adopted within the holochain framework as a design requirement for indigenous sovereignty

²Pepeha is a method of self-introduction in māori. It reveals your identity by describing your relationships with significant people and places. [154]

6.2.1 Understanding Indigenous Data Sovereignty Principles

Throughout our efforts to create a Holochain Application that reflects the sovereignty of indigenous data, we gained a profound appreciation for the close connection between data and the cultural identity, spiritual convictions and customary rituals of indigenous groups. Our approach to development was significantly shaped by recognizing the vital role of indigenous data in these communities. This comprehension guided the development and implementation of our holochain application with the goal of improving the representation and empowerment of indigenous communities.

We recognize that simply adopting the principles of indigenous data sovereignty in theory was only the first step. It was essential for us to take concrete actions to embed these principles into the technological foundation of our prototype. Furthermore, we structured the governance of our holochain application to mirror the inclusive decision-making processes typical in indigenous communities, ensuring that their perspectives were actively incorporated in the management and development of the technology. The protection of indigenous data sovereignty within our holochain application required a comprehensive approach focused on security and resilience. This included protecting the integrity and confidentiality of indigenous data from external threats and unauthorized access. Although security remains an ongoing discussion within holochain communities, we strived to implement feasible measures without hindering the overall progress of the application. In summary, our initiative to integrate indigenous data sovereignty principles into the creation and deployment of our prototype involved acknowledging the importance of indigenous data and building trust and confidence in the protection of indigenous knowledge.

6.2.2 Applying Indigenous Data Sovereignty Principles

In our effort to integrate the principles of indigenous data sovereignty into the holochain platform, we designed access control features that adhere to these principles. We prioritized concepts such as ownership, which align with the robust features of the Rust programming language, ensuring that the holochain framework concretely and significantly embodies indigenous data sovereignty. The creation of these access control features provided agents within a holochain network with greater confidence that their data is secure and localized. The creation and deployment of these access control mechanisms required extensive research into both holochain and its current capabilities. It became clear that meaningful collaboration with indigenous communities was the optimal approach to advancing our phase and would have eased our workload, though time constraints prevented this. Nevertheless, we ensured our access control mechanisms could adapt to the evolving needs and goals of indigenous communities. This collaborative effort not only supports the sovereignty and self-governance of indigenous communities but also sets a standard for developing inclusive and ethical technology. Ultimately, incorporating the principles of indigenous data sovereignty into access control mechanisms shows a commitment to respect, empowerment, and collaborative progress in the tech sector, fostering a more equitable and inclusive digital future for cultural communities. To evaluate the effectiveness of our holochain prototype application in reflecting indigenous data sovereignty principles, we considered

several key factors, including:

- **Data Ownership and Control:**
The objective is for the artifact to ensure that indigenous communities have complete ownership and control over their data. This entails their ability to determine the individuals who can access the data, the manner in which it is used, and the intended purposes for its use.
- **User Experience:**
The focus of the artifact should be on enhancing user experience and facilitating the interaction and management of data for indigenous communities. This can be achieved by creating intuitive and user-friendly interfaces, providing clear instructions and guidance, and providing support services to address technical challenges.
- **Security and Privacy:**
To ensure the confidentiality, integrity, and availability of indigenous data, it is important that the artifact implements robust security measures. These measures may involve the use of encryption protocols, access controls, and periodic security audits to detect and resolve vulnerabilities.
- **Cultural Sensitivity:**
The artifact should exhibit cultural sensitivity by integrating indigenous protocols, languages, and values. This can include offering choices for language preferences, integrating cultural symbols and artwork into the user interface, and guaranteeing that data practices align with indigenous values of honoring cultural protocols and traditional knowledge.
- **Inclusivity:**
The aim is to ensure that the artifact is easily accessible to indigenous communities, considering aspects such as the availability of Internet connection, compatibility with various devices, and overcoming language barriers.
- **Scalability:**
Scalability is an important aspect to consider when designing the artifact, as it should be able to accommodate the increasing needs and demands of indigenous communities. Taking these factors into account, a comprehensive and inclusive approach to data governance can be achieved.

Considering the key factors mentioned, we treated these as artifacts and integrated them as features for the prototype. Additionally, we ensured to re-prioritize if one feature took longer than another within the given timeframe, ensuring overall progress. Our documentation process was thorough enough to highlight the challenges and successes we would later describe. By detailing these aspects, we aimed to provide a clear picture of what was time-consuming and beneficial compared to the artifacts that needed more extensive iterative improvements. In our efforts to create a holochain application based on the previous iwi application, we followed a series of essential steps. Additionally, we aimed for the second version of our prototype to feature a flexible and adaptable design and architecture, capable

of effortlessly supporting various data governance models. This strategy guarantees that indigenous communities maintain control over who can access and use their data.

Throughout this phase's development, we prioritized the inclusion of indigenous practices. One method we employed was incorporating indigenous languages into the platform's user interface, making it more welcoming and accessible for indigenous users. Additionally, we deemed it crucial to establish strong mechanisms for accountability and transparency within the holochain ecosystem. Our experiment encountered challenges and setbacks, highlighting the complexity of integrating these nuanced considerations. Despite these obstacles, our efforts represent a committed attempt to respect and integrate indigenous data sovereignty in a way that honors and uplifts indigenous communities and their rights. It is debatable whether the accountability and transparency mechanisms within the holochain ecosystem are as robust as claimed. The challenges and setbacks experienced during the experiment revealed deeper issues within the system that could impede its ability to genuinely honor indigenous data sovereignty in its current form.

6.3 Building A Holochain Application - Final Version

As mentioned earlier, this particular version includes several changes inspired by open source repositories that contribute to the holochain development framework³, to achieve the objective of improving our application by developing a profile similar to centralized identity systems containing user information [150]. These modifications encompass attributing agent functions. We had to make several adjustments to the holochain application in order to align it more closely with a guided indigenous data sovereignty framework. However, due to time limitations we were only able to make significant changes that somewhat resembled a cultural setting. Additionally, we were able to showcase these modifications by implementing the aforementioned guided principles. These changes were our best effort to highlight the challenges faced when developing a holochain application from an indigenous data sovereignty perspective, as well as the common mistakes encountered throughout this process. We showcase these adjustments in this section. In alignment with our methods, we conducted the same tests as detailed in 8. We implemented changes akin to a *Pepeha* [154] and made structural modifications to the integrity zone of our application to ensure immutability, with all changes being managed under version control. Table 6.2 lists the New Zealand regions used in our profile descriptions to indicate origins. This concept is inspired by the traditional māori practice of introducing oneself based on one's place of origin. The pepeha is increasingly adopted as a form of introduction in the public sector and political arenas, as it utilizes the māori language, which is a taonga (treasure) and an essential aspect of māori culture. We decided to adopt māori indigenous sovereignty practices for our second application due to the significant connections and influence of the contributors to this paper.

³<https://github.com/holochain-open-dev/profiles>

No Hea Koe	Maunga	Moana	Maunga
CapeReinga	Tutamoe	Ninety Mile	Huka
Russell	Tararua	Doubtless	Waikaremoana
Hokianga	Moehau	Rainbow	Rere
Whāngarei	CastleRock	Whangarei	Ruakituri
Dargaville	Rangitoto	Kaiwi	Mount Damper
Auckland	Wellington	Kaipara	Whanganui
Waiheke	Eden	Omaha	Waihi
Thames	Hobson	Orewa	Taupō
Huntly	One Tree Hill	Long	Rotopounamu
Ngāruawāhia	Albert	Hauraki	Emerald
Morrinsville	Roskill	Onetangi	Taranaki
Hamilton	Māngere	Muriwai	Waipunga
Tauranga	Bluff Hill	Pupuke	Shine
Cambridge	Te Toiokawharu	Takapuna	Waiau
Tokoroa	Kohukohunui	Watematā	Tutaekuri
Rotorua	Te Aroha	Cheltenham	Patea
Whakatāne	Maunganui	Mission	Moawhango
Gisborne	Karioi	Piha	Turakina
Tolaga Bay	Pirongia	Karekare	Whenuakura
Wairoa	Kakepuku	Manukau	Tukituki
Taupō	Manguatautari	Hunua	Manawatu
New Plymouth	Ngongotahā	Cathedral	Hutt
Hāwera	Tarawera Dome	Hot Water	Ruamahanga
Wanganui	Edgecumbe	Pauanui	-
Napier	Pureora	Whangamata	-
Hastings	Tauhara	Owharoa	-
Taranaki	Tongariro	Waikato	-
Feilding	Te Heuheu	Kaituna	-
Palmerston North	Ruapehu	Waitoa	-
Dannevirke	Pouakai	Ngarunui	-
Levin	Taranaki	Bridal Veil	-
Masterton	Te Mata	Rotoiti	-
Wellington	Kahuranaki	Rotorua	-
Featherston	Hikurangi	Tarawera	-
Upper Hutt	Mangaweka	Ohope	-
Lower Hutt	Rimutaka	Marokopa	-
Porirua	Colonial Knob	Whakatāne	-
-	Kaukau	Wairere	-
-	Te Ahumairangi	Omaru	-
-	Victoria	Ongarue	-

Table 6.2: New Zealand Geographic Region Used In Profile Setup

Alongside the profile regions selected at the application’s initiation, we have introduced and expanded the entry-types to encompass a collection that includes health records. We chose to display health records based on the belief that social communities with close-knit trust and principles aligned with 6.1, such as transparency and trust, may have a high degree of trust within their communities and whanau. Consequently, records within this distributed network may not face issues of vulnerability and privacy and may, in fact, be shared among members to foster greater unity and compassion. However, we took a dual approach: to investigate how health records might function on a distributed system and to identify privacy concerns that may arise if a technology like holochain is utilized for its potential benefits. Here, we outline the definitions for our zome integrity types as referenced in 14.

Procedure 14: V2 Application Integrity Types

Definition:

PatientRecord refers to the patient record within indigenous communities.

Require: hdi

```

1 enum Indigenous {
2   Maori,
3   Other,
4 }
5 macros ← [hdk_entry_helper, derive(Clone, PartialEq)]
6 pub struct Comment {
7   content_comment ← type ← String,
8   patient_record_hash ← type ← ActionHash,
9 }
10 macros ← [hdk_entry_helper, derive(Clone, PartialEq)]
11 pub struct PatientRecord {
12   content ← type ← String,
13   resource_type ← type ← String,
14   date_visited ← type ← Timestamp,
15   whanau ← type ← String,
16   ingoa ← type ← String,
17   no_hea_koe ← type ← String,
18   maunga ← type ← String,
19   moana ← type ← String,
20 }

```

Observe that *PatientRecord* contained values that matched those of a Mori *Pepeha*. This allowed us to incorporate a cultural taonga (treasure) into the application’s respective modality, achieving our aim of merging aspects of culture with technology. Subsequently, we focused on the library functionality to implement and finalize helper functions necessary to bridge the gap between the familiarity of a modern accessible function like sorting and the holochain’s application. We emphasize some of these features based on the naming convention shown in the following 15.

Procedure 15: V2 Application Entry and Link Types

Definition:

PatientRecord refers to the patient record within indigenous communities. *Comment* refers to the notes connected to a *PatientRecord*

Require: hdi, comment, patient_record

```

1 macros ← [derive(Clone, PartialEq), serde(tag =
   "type"), hdk_entry_defs, unit_enum(UnitEntryTypes)],
2 pub enum EntryTypes {
3   PatientRecord (PatientRecord),
4   Comment (Comment),
5 }
6 macros ← [derive(Clone, PartialEq), hdk_link_types]
7 pub enum LinkTypes {
8   PatientRecordUpdates,
9   PatientRecordToComments,
10  CommentUpdates,
11  AllRecords,
12  RecordsByRecorder,
13  RecorderToRecords,
14  RecordsToRecorder,
15 }

```

As seen with the *LinkTypes*, we added features based on their naming convention, such as *RecordsByRecorder*, which filters all records created by the author, known as the *Recorder*. The same logic applies to the other *LinkTypes*. These types were additional features incorporated into the new prototype. We believed that an application that feels more familiar to users would be adopted more easily. Building on what has been demonstrated in this section, we emphasized several features that complemented each other. However, these features required corresponding UI components to enhance the design, ensuring that both the discussed features and the additional UI elements were effectively integrated.

6.3.1 IDSOV V2 UI Design

With the advancements in UI technology, we incorporated supporting libraries into the project that added significant value. These components simplified the development of front-end clients compared to the standard components provided by the holochain scaffolding tool. The libraries DaisyUI [140] and Tailwindcss [156] are contemporary and well established, playing a crucial role in accelerating our front-end development. These packages were added to the *idsov-hApp* submodule's *package.json* directory. The remaining packages were supplementary from the Svelte UI framework we selected. When comparing the previous version of our application with the current one, it was clear that a

modern look was necessary. The design 6.2 of the application reflects our ambition to blend current trends with innovation.

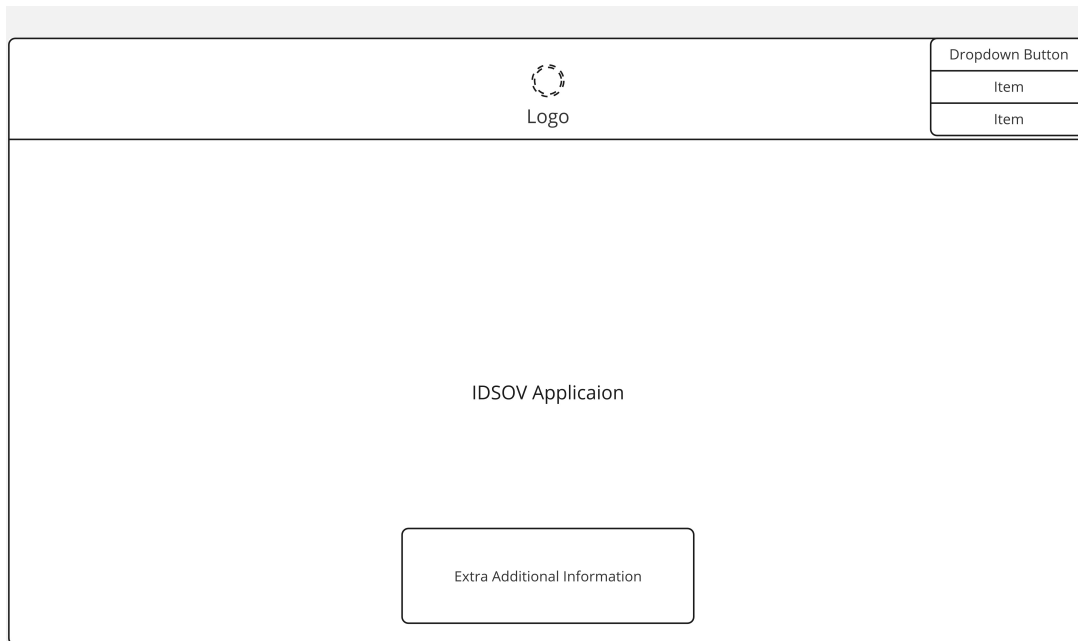


Figure 6.2: IDSOV V2 Landing Page Frame Design

With the newly integrated library support, we managed to implement the design with extra UI features such as a dark-mode component, logo integration, and some appealing animations that are only visible when the application is running. Although these features were not part of our artifact selection, the support of *DaisyUI* and *Tailwindcss* made this implementation of UI features simple and effortless. The landing component of the application is shown here 6.3. In addition to the frame designs and live UI application, we also showcase the combination of cultural activity within the record management system that we built within our holochain network. These designs reflect our commitment to incorporating indigenous values and data governance to a distributed application which we have not seen yet within the holochain community for a patient-like system record. It can be argued that while integrating cultural activities and indigenous values into a record management system within the holochain could appear innovative, it introduces numerous unknown security issues that could potentially complicate the system and reduce its user-friendliness and desirability [74]. The main priority of a record management system should be efficiency, user-friendliness, and data security rather than cultural elements that might not necessarily improve the system's functionality. Furthermore, adding too many unique features could create compatibility problems with existing external systems and impede widespread adoption within the holochain community.

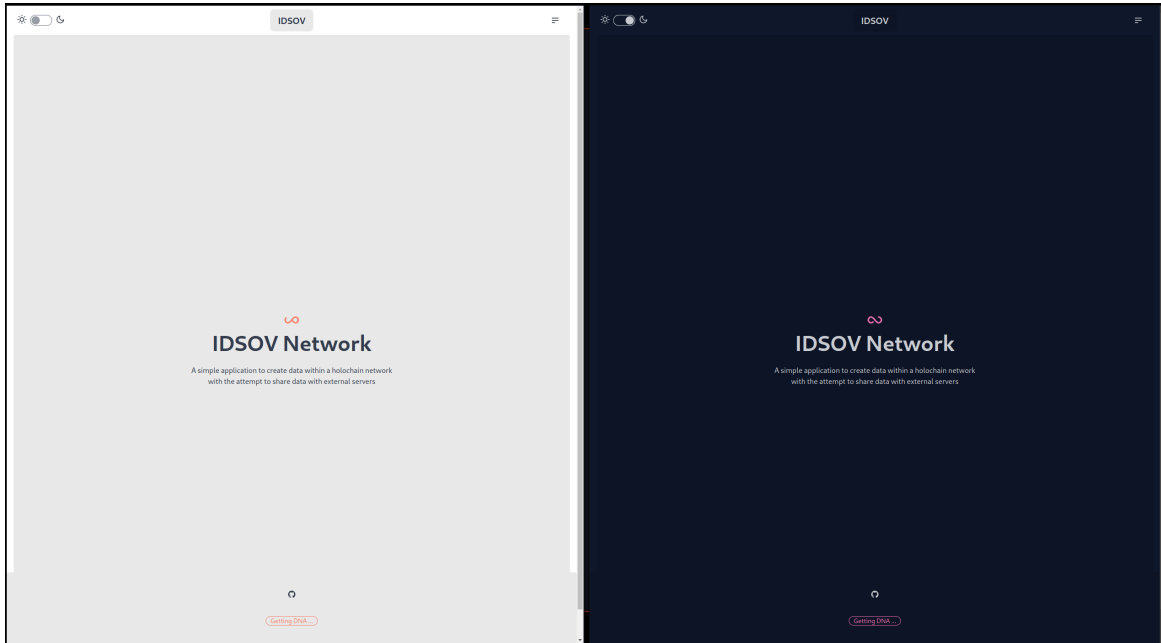


Figure 6.3: IDSOV V2 Landing Page Live Application - Dark and Light Mode

In our landing page application, we chose to design a user interface that features a drop-down box illustrating the stages of our prototype and the overall outcome, along with a registration option. The registration option enabled users to create a profile within the application. Once registered with an agent profile, users are granted access to the application’s dashboard via a routing mechanism in the UI framework *Svelte*. The dashboard component is where the records, defined by our integrity type definitions and coordinator zones, are utilized. Each component mirrors the core elements of our zones related to the patient record, which is just one part of the DNA. Another component is the comment section, which contains notes for each patient record. Although these features are interconnected, it was crucial for us to separate them to ensure best practices, with each service serving its specific purpose. In the following section, we showcase the defined entry types in action through the UI. We believe that incorporating additional UI elements into our landing page and overall design was crucial for providing users with an optimal experience. Studies [112] emphasize the importance of information design, interaction design, and interface layout, which we aimed to achieve through our user-centric design approach.

6.3.2 IDSOV V2 Integrity Types Display

By incorporating additional front-end libraries, we were able to enhance the flexibility of the records through spatial input boxes, color gradient selection, and a dark mode feature. Initially, our implementa-

tion faced challenges due to compatibility issues with rendering and routing in the holochain client-side packages. This necessitated finding solutions that were both simple to implement and time-efficient. Consequently, we discovered that *DaisyUI* and *Tailwindcss* were the most straightforward options, thanks to their seamless integration with native components already utilized by the holochain client framework, such as *Svelte*, which allowed us to concentrate more on implementation rather than bug fixing. The packages for viewing are located in the repository ⁴ inside the submodule *idsov*, located in the root of the parent directory.

After the user logs into the application with their profile details, they receive a notification confirming successful sign-up. They are then given additional information on how to create patient records, which are distinct from those of the *recorder* who registered. This distinction is because the user who signed up is considered the agent and author of the data within the network, and other agents might also have this capability. While there are numerous ways to enhance this, we focused on the core concepts we wanted to highlight. We aimed to expand and refine elements related to the call to action [105] concerning data utilization within the health information system by incorporating features like filtering, general dashboard functionalities, and profile management within the application.

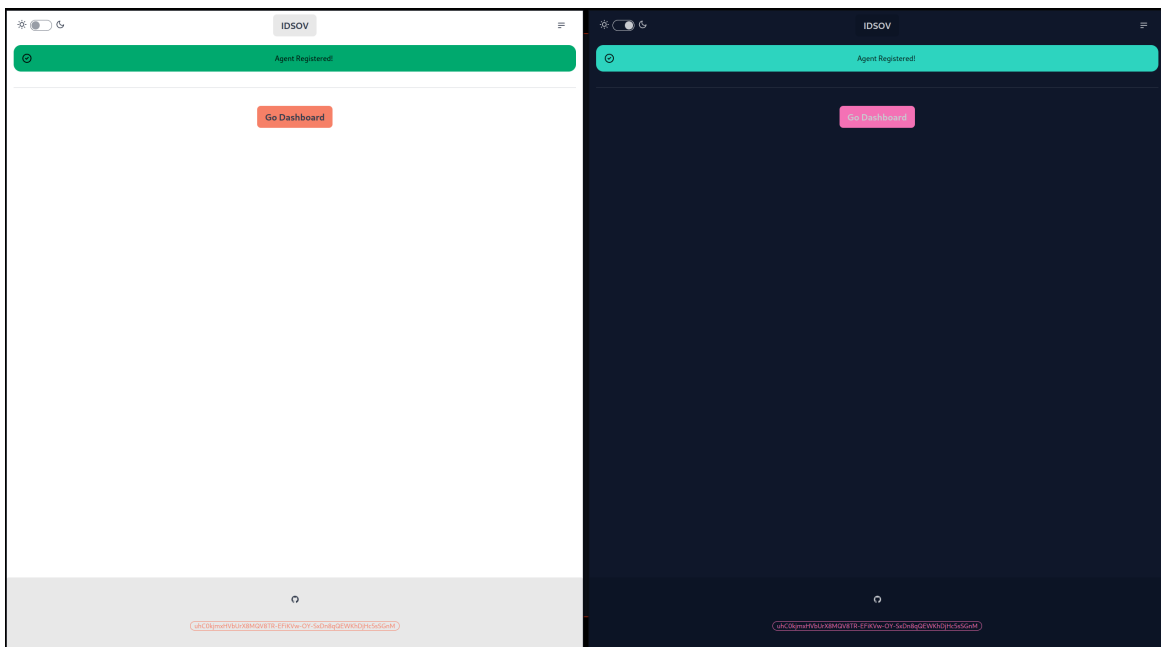


Figure 6.4: IDSOV V2 Successful registration of two agents

In this section, we showcase several features derived from the integrity zome and entry-types previously discussed 14. The user interface includes a shared dashboard feature for all *patient_records*

⁴<https://github.com/onahp/paradigm-app>

created by all agents. These records are public rather than private. Although we had the option to mark these entries as private for each agent, we decided it would be more advantageous to highlight the various features familiar to users. However, we do provide a privacy-like filter feature through embedded links that illustrate what a list of private entries might look like on a dashboard. We call this UI element *My Dashboard*. This tab displays all entries created by the agent as shown 6.5.

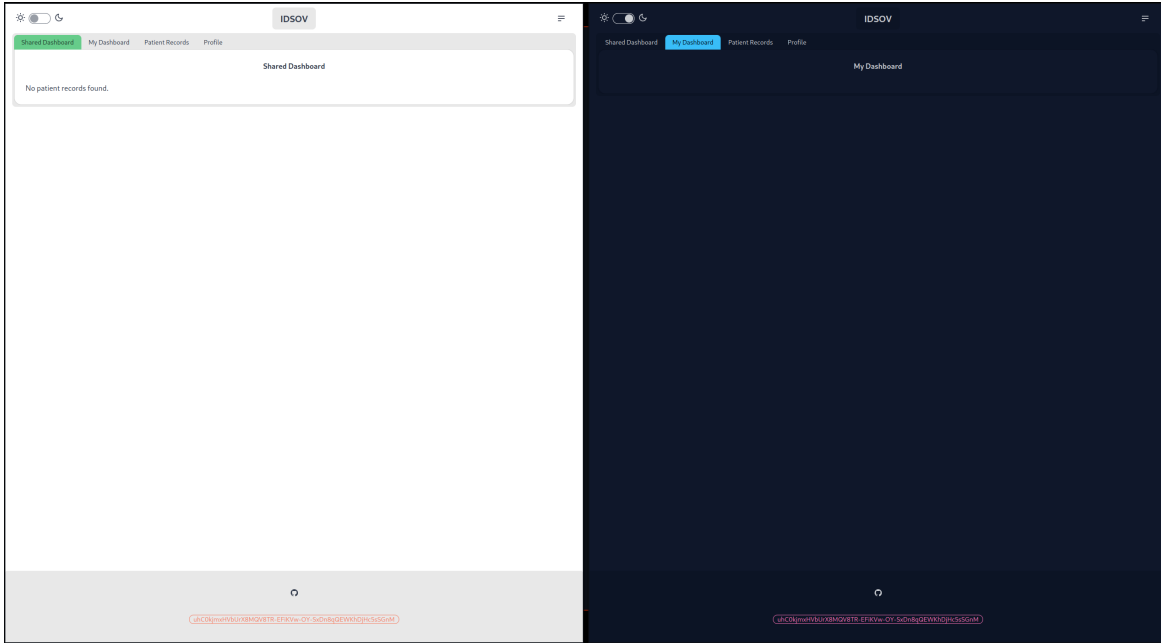


Figure 6.5: IDSOV V2 UI dashboard

We present 6.6 the patient record functionality within the application alongside the profile list feature implemented⁵ that displays the number of agents on the network. Whenever an agent creates a record, it is propagated to both the general dashboard and the agent’s dashboard. In the following section, we delve deeper into the internal mechanisms and discuss the significance of these features in our efforts to discover whether holochain is a relevant technology that supports indigenous sovereignty.

6.3.3 IDSOV V2 Functionality

In our application, we utilized *links* that are part of the holochain application, to help create connections between records. An example 16 of this is what we did to create *links* between *patient_records* and their respective agent. We achieved this by first creating a path that is inherently added as the entry hash within a link. Paths denote a single journey through a tree structure to a specific point. The

⁵<https://github.com/holochain-open-dev>

primary goal is to enable recursive traversal back up the tree, performing hashing, committing, and linking of each sub-path until the root is reached [153]. This allows us to establish a path called *all_records*, include the path as the entry hash in the link parameter, with the target hash being the patient record generated, and ultimately utilize the tag from one of the public enum types we designed named *AllRecords*. Other link types are applied according to their respective naming conventions.

Procedure 16: V2 Application Path With Link Usage

Definition:

The *path* is a struct that takes a vector of components. A *component* is also a struct that takes a vector of **u8** values made of arbitrary values that are to be hashed in a predictable way used for traversal.

Require: *hdk*, *patient_records_integrity*

- 1 *path* ← *Path*("all_records")
 - 2 *create_link* (*path*, *patient_record_hash*, *LinkTypes::AllRecords*)
 - 3 *Get all records*
 - 4 *get_links* (*path.path_entry_hash*()?, *LinkTypes::AllRecords*, *None*)
-

Consequently, when we create multiple records, we have these records associated with the *all_records* path and the *AllRecords* tag in a link, which can be used at any moment to access all created records. This allows the identification of records for indigenous communities to mirror modern technology, aiming to achieve comparable results with contemporary record management systems. We acknowledged that using *anchors* was an option, as they are akin to links in that they allow us to connect entries. However, technically, the anchor entry definition serves as the path definition [138]. For implementation purposes, we chose to use links because they fundamentally align better with our objectives for managing and understanding entries in the context of indigenous data sovereignty. We present a list of functionalities identical to 16 in the following table 6.3.

LinkTypes	Functionality
<i>PatientRecordUpdates</i>	Traverse updated records
<i>PatientRecordToComments</i>	Collect Comments under a patient record
<i>CommentUpdates</i>	Traverse comments responded to
<i>AllRecords</i>	Traverse all records
<i>RecordsByRecorder</i>	Traverse records by author

Table 6.3: LinkTypes to artefacts

In our attempts to develop useful features for the IDSOV application, it became clear that using these *LinkTypes* for path entry retrieval was beneficial for labeling custom functionalities for indigenous communities that need specific data points according to naming conventions. The *LinkTypes* use case could address the unique needs of communities, similar to traditional filter-like retrieval for user-specific

requirements. This was our main source for creating artefacts and determining which features were available in the zones. Additionally, the HDK suggests the 80/20 rule, where 80% of the application can be production-ready using 20% of the HDK's high-level features. Upon closer examination of the outputs within the source-chain, it became evident that the source chain included the *genesis records*, which are the four elements created when an agent is initially set up to join the holochain DHT which we were able to successfully build with our app. Furthermore, during the setup of our application, we generated a new sample patient record by one of the sample agents, Bob 6.7. This record demonstrated our predefined enums from the integrity zones established for the patient record. We observed that the source chain contained the entry content of our sample record. One of the features we had wished to integrate into our application was a *capabilities* module, this is demonstrated by Poor [105] on how capabilities in action are displayed. We talk a little bit about why this was a key artefact we would have wished to administer.

Capability claims and grants are functions that can be accessed within an agent's cell. Grants are system output entries recorded on the source chain that specify access permissions, while claims are system output entries that refer to a grant on the source chain. There are four different types of capability grants, each with a brief explanation [142].

1. Author

The *Author* of the local source chain provides their agent key as a claim and has complete access to all functionalities. This is similar to having administrator privileges as a local user.

2. Unrestricted

The *Unrestricted* access means that any individual can invoke this function without needing to present any form of claim. This is akin to having no security measures in place, permitting any agent to make calls without requiring any proof.

3. Unassigned

The *Unassigned* access allows individuals who possess the randomly generated secret key from a grant to invoke this function.

4. Assigned

The *Assigned* grant permits designated agents to invoke this function, provided they supply the corresponding secret key.

Consider that indigenous communities might appoint trusted individuals, such as family members or reliable community members, to carry out actions on their behalf. In this scenario, capability grants would be a beneficial feature of this technology. This aligns with some of the core principles 6.1, embedding trust, dignity, and privacy into the technology to honor indigenous rights [116]. Therefore, it was worth mentioning in this section, even though it was not implemented as a functional module in the application.

One of the features we implemented was a simple function to display the hash value of the DNA you are part of [143]. This proved to be beneficial for users to verify if they are in the same DHT

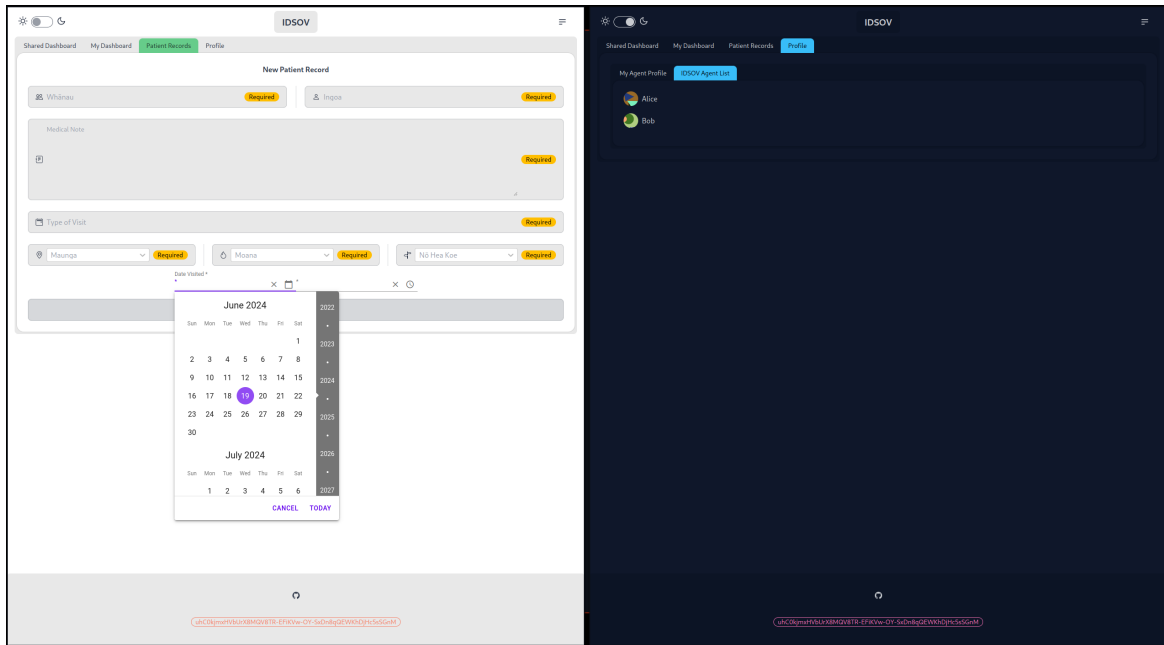


Figure 6.6: IDSOV V2 UI patient record with agent list

space as other known participants, assuming mutual awareness among participants. This minor feature, which we placed at the very bottom of the page as shown in figures 6.6 and 6.7, enables users to view the DNA value, providing contextual information from the host. This feature is provided by the HDK and can be easily implemented by the developers who wish to have this available for agents. Here 17, we present our simple implementation and explain its significance for indigenous communities. The DNA values are unique to the members of the DHT space, this should not be mistaken as the agent information but the DNA information. Indigenous communities might leverage this integrated feature to establish a space showcasing an official DNA record, allowing all participants to confirm the space’s legitimacy. Initially, we incorporated this feature within the coordinator zome of the patient records; however, upon review, it could be more appropriately placed in a different library component for improved readability and best practices.

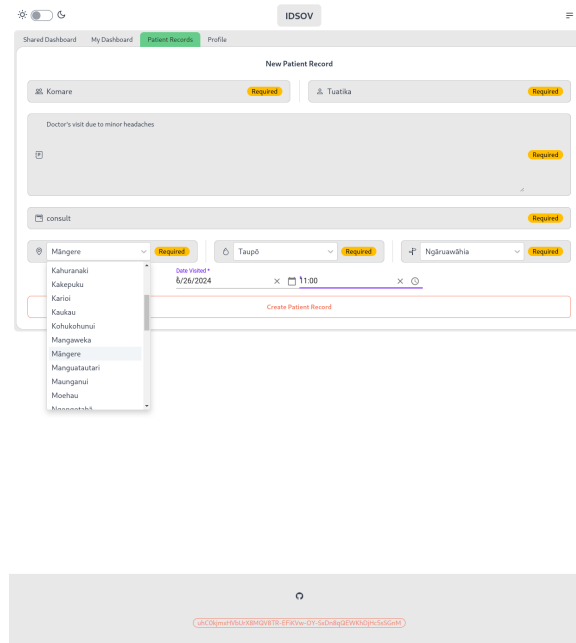


Figure 6.7: IDSOV V2 UI creating a patient record

Procedure 17: V2 Application DNA Info Implementation

Definition:

The *dna_info* function does not require any parameters and is imported from the *hdk::info*

Require: *hdk*

```

1 macros ← [hdk_extern]
2 get_dna_hash (.: ()) → ExternResult < String > {
3   dna_value ← hdk::info::dna_info()?;
4   Ok (dna_value.hash.to_string()),
5 }

```

As discussed in this section and the preceding subsections, we developed a successful application that incorporated several key features with standard CRUD operations. Additionally, we were able to add links that acted as filter-like artifacts to display records within the application. Finally, we presented a built-in feature of holochain that displays the DNA value of the space in which the user is participating. The second iteration of the application integrates insights from the holochain organization's sample setup and our initial version. We utilized the scaffolding tool to create the first version and then incorporated our learnings from both experiences to develop the final version two.

The application is available as a submodule in the repository, accessible here ⁶.

6.4 Challenges and Findings

During the development of a holochain application focused on indigenous data sovereignty, we faced several challenges that need to be addressed to create a more viable product and solution. One such challenge was the latency in information reception between agents. While integrating our first and second versions of the application, we observed significant downtime in data reception among peers. This issue might stem from the background validation logic and signaling, but we discovered that frequently refreshing the application enabled us to see the newly shared data between peers. One of the challenges we faced was the availability of documentation for advanced features. During implementation, we had to rely on information from the Rust HDK to understand how to build our functions and their internal workings. This was somewhat different from the official documentation, which mostly contained high-level abstractions and examples for CRUD operations. This was a disadvantage because we wanted to push the limits of what holochain could offer in its advanced stage for one of our implementations. Although we acknowledge that time constraints were a factor, we found that the documentation was generally an issue for us. A further problem we faced was the limited number of front-end frameworks displayed with holochain, with only three currently supported as listed on their website. The reason for this limitation is unclear, but it is plausible that the restricted choices, combined with insufficient documentation, could deter developers from creating effective solutions for indigenous data sovereignty. The composition of data modeling and domain logic in holochain can be somewhat challenging due to the lack of flexibility in operational methods and lack of samples for use cases. Instead, there are prescribed practices that are recommended to follow, which are beneficial for understanding the concepts and their application. While this provides a good foundation, it becomes more complex when delving into advanced topics. Overcoming these obstacles would facilitate the implementation of an inclusive strategy, aiding indigenous communities in understanding how the technology can serve as an effective means for regaining control over their data.

6.5 Evaluation

Creating an application while the holochain documentation was still evolving presented some difficulties. However, most basic CRUD operations were straightforward to implement thanks to the scaffolding tool. This tool was instrumental in the initial setup, and we anticipate that more features will continue to develop and mature over time. Essentially, the technology we utilized allowed us to explore new ways to support indigenous sovereignty by managing their data locally and preserving it among participants within the DHT space. Our experience indicates that while holochain has its specific challenges, it holds significant potential as a viable solution.

⁶<https://github.com/onahp/paradigm-app>

The task of developing an application with a limited set of mature features in this prototype was challenging, as mentioned above. However, there were some pleasantly supported features in the holochain ecosystem, thanks to the robustness of Rust and other mature technologies. These technologies enabled certain support to be enabled within the holochain ecosystem.

Chapter 7

Research Phase 3: Shared paradigm

In our concluding phase, we investigated the potential for data sharing between holochain and external service providers. This journey takes us back to our initial phase, where we set up a centralized data management server, reflective of contemporary platforms that store data. We combined this with insights from our second phase, which examined the use of holochain applications as a feasible solution for indigenous data sovereignty. Observing that most modern data platforms operate on the cloud, we recognized on-premise servers as another viable option. Typically, this is seen as a hybrid approach in many practices. Thus, we explored whether a hybrid model—utilizing cloud-based computing power alongside a holochain application with self-hosted capabilities and peer-to-peer sharing—could be a viable strategy for a shared paradigm. This investigation led us to several challenges and insights that we present in this section.

7.1 Goals and Objectives

This section was driven by a singular objective. Considering that we developed a range of services hosted on AWS, featuring data sharing, and a holochain application providing peer-to-peer functionalities, such as entry validation and CRUD-like operations within the application, our objective was:

- Can data be exchanged between external providers and holochain?
- How is data shared in this context?
- Is this a feasible solution for indigenous communities aiming to utilize cloud computing capabilities while maintaining data sovereignty, as in the case of holochain?

7.2 Establishing A Connection

In our effort to establish a connection for the platform, we examined the possibilities of using HTTPS and Websockets. Our initial approach was to experiment with Websockets, not in any specific sequence, but because holochain also employs Websockets. We decided to create two client Websocket applications in the programming languages commonly used by holochain: JavaScript and Rust. This choice was made to align closely with holochain’s technology while avoiding the need to learn a new language solely for creating Websocket and HTTP requests, allowing us to concentrate on developing client functionalities. During the creation of our client websocket applications, we specifically chose to develop these applications independently from the holochain environment. The environment was not hosted on instances provided by cloud providers, nor was it integrated within the application as an additional client inside the framework. Instead, this was a standalone program running on a machine, making initial calls to the cloud provider.

Our rust client used two library packages: *serde_json* and *websocket*. We started our application by creating a main function to instantiate it and then called the appropriate websocket listener hosted on the cloud. We transmitted a payload containing our intended action along with a message payload. After the listener received our message, we expected a reply to confirm a successful connection. Below is a demonstration of our rust websocket client.

Procedure 18: Rust Websocket Client Listener

Definition:

We characterize our rust client as using an unencrypted connection to lay the groundwork for communication between the cloud and the client.

Require: *serde_json*, *websocket*

```

1 fn main() → Result < () > {
2   client ← ClientBuilder :: new(websocketlink)
3     .unwrap()
4     .connect_unsecure()
5     .unwrap();
6   payload ← {action : "sendmessage", data : "message"}
7   client_message ← serde_json(payload)
8   client.send_message(client_message)
9   response ← client.recv_message().unwrap()
10  println! (response)
11 }
```

We managed to establish a successful connection to our listener hosted on the AWS cloud platform, confirming that the listener operated as anticipated from our phase one setup. Initially, our goal for the Rust client was to determine if the backend host in holochain could interact with external service providers outside of the DHT environment. This was also the initial hypothesis for the WebSocket

client developed in TypeScript.

Procedure 19: Typescript Websocket Client Listener

Definition:

We characterize our typescript client as using an unencrypted connection to lay the groundwork for communication between the cloud and the client.

Require: websocket-ts

```

1 websocketConnection ← WebSocketBuilder(wsslink)
2   .withBuffer(ArrayQueue ())
3   .withBackoff(ConstantBackoff (2000)).build()
4 message ← {action : "sendMessage", data : "message"}
5 websocketConnection.send(message)
6 websocketConnection.addEventListener(
7   WebSocketEvent.open, () → log("open")
8 )
9 websocketConnection.addEventListener(
10  WebSocketEvent.error, () → log("error")
11 )
12 websocketConnection.addEventListener(
13  WebSocketEvent.reconnect, () → log("reconnected")
14 )
15 websocketConnection.addEventListener(
16  WebSocketEvent.close, () → log("closed")
17 )
18 websocketConnection.addEventListener(
19  WebSocketEvent.message, responseFunction()
20 )

```

When the rust and typescript clients were executed, initial connections with the listener were established. However, despite these initial attempts, we encountered issues in forming a stable connection. The problems faced with our client connections to the listener were multifaceted. Conducting the experiment in a home setting on a Linux machine instead of a controlled lab environment with stable Internet led to occasional disconnects and disconnections. Additionally, shared CPU resources and limited RAM usage, due to other concurrently running programs, contributed to the issues. Despite these challenges, we document our experiences and discuss them in the next section. During our experiment, we did not focus on transmitting a particular payload, as our main goal was to establish a connection, which we did successfully. Although this marked the experiment as a success, it did not meet our main objective of evaluating the data-sharing capabilities between holochain and a modern platform for indigenous sovereignty, leading us to explore further alternatives.

Since our client scripts successfully established a connection, the logical next step was to integrate them into the holochain application we were developing. During the initial implementation of the websocket client written in Rust, it became evident that this approach was unfeasible due to the holochain's architecture involving its WASM component and conductor services, which made it difficult to connect to our centralized platform. The challenges encountered in connecting to AWS from within holochain underscored the need to explore the network layers and architecture more thoroughly. This scenario also led to reflections on the holochain framework's capability to support external cloud services, which are often crucial for building and deploying scalable applications and the multitude of integrations that many companies and people use today. Ultimately, this emphasized the importance of adaptability and versatility in the architecture of decentralized frameworks and applications, particularly when integrating with popular and essential services like the major cloud providers Azure, AWS, and Google.

Following that experience, we moved onto the typescript client as the holochain client sits outside of the back-end components of the holochain application, and we were able to establish a connection based on a separate experiment that confirmed this for us. The experiment did not involve running websocket clients, but instead making simple HTTP calls from within the framework on the client side of the holochain application to external service providers. We added code to the *AllRecords.svelte* file and created a simple function that requested the following URL¹. JSONPlaceholder provides a dependable mock API that is free for testing and prototyping purposes [151]. The responses to our simple requests are detailed in the procedure 20 and are demonstrated within the application console² as illustrated in 7.1.

Procedure 20: Holochain Client External HTTP Call

```

1 response ← fetch(URL)
2 data ← await response
3 log (data)

```

Considering that the holochain application can make external calls from the client side, it opened up opportunities for us to explore the feasibility of a client-side workaround versus integration with an external provider. However, this approach carries significant security risks and vulnerability to client-side attacks, making the use of client-side storage for API usage not cost-effective [20], [93]. This led us to dismiss the idea of storing API keys on the client-side. Nevertheless, the potential is not completely disregarded, and further investigation into the security implications of a client-side holochain might be worthwhile, as our assessment was primarily based on web client-side literature.

¹<https://jsonplaceholder.typicode.com/posts>

²The data shown in the console from the HTTP request is retrieved from an external provider, and no code was written by us to display it on the dashboard. We only show the data in the console to demonstrate that the request was successful.

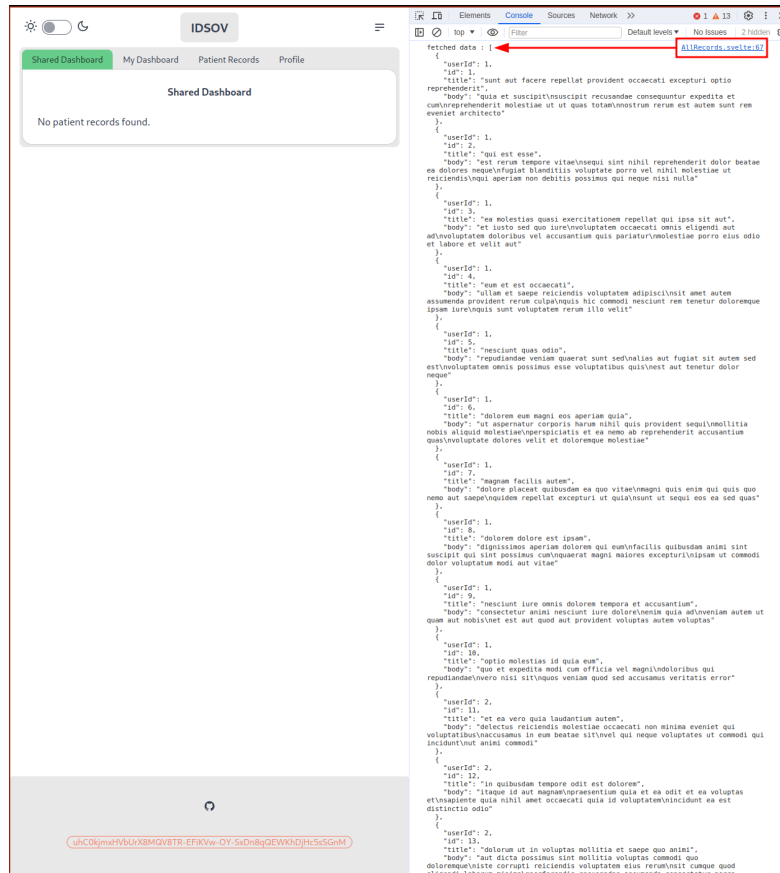
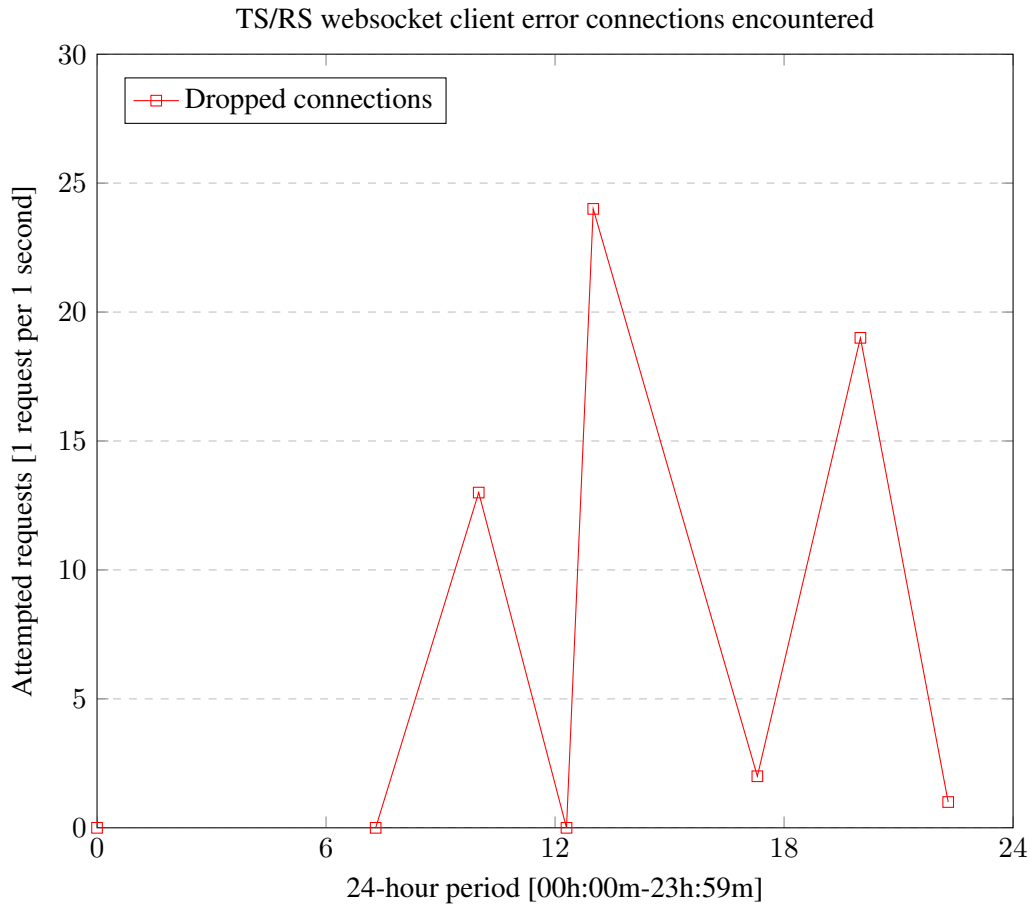


Figure 7.1: Holochain Client External HTTP Call in Application Console.

7.3 Challenges and Findings

During this phase of our experiment, we faced several challenges. A minor problem was the instability of client connections between Rust and TypeScript clients and the cloud provider at times 7.3. Although our experiment was not performed within the holochain client-server repository, the unstable connections highlighted potential problems that might arise if our infrastructure or client configuration were inadequately established. However, we recognize that this issue might have occurred due to various factors, including the reliability of our local ISP connection, the initial configuration of our websocket clients, or the establishment of our infrastructure outside of a lab environment with stable high available resources. At the beginning of phase three, we faced another challenge in which the SST framework, crucial to our infrastructure development, introduced breaking changes [152]. These changes necessitated significant refactoring, which stalled our phase three development and

subsequently delayed our planned progress. The procedures from phase one that formed the foundation of our infrastructure are still in place. However, there are some notable differences in formatting and structuring. These differences are significant enough to be acknowledged, but not substantial enough to necessitate a complete overhaul of the procedural stacks should we continue to use SST V2 over the breaking changes introduced for SST V3.



The limitations of these constraints were especially noticeable in the third phase of our investigation, which focused on establishing data-sharing functions between a holochain application and a centralized platform. The lack of in-built support to establish websocket connections with outside services highlights a crucial obstacle to interoperability that may restrict the effectiveness of holochain in situations where integration with conventional web services and structures is necessary. This discovery is crucial for developers and researchers operating in the holochain environment, as it emphasizes a substantial constraint that needs to be resolved to enhance the framework’s usefulness and acceptance in various applications and services.

Despite these constraints, the open-source aspect of the holochain project offers a distinctive

chance for the community and developers to collectively tackle these issues. Adapting or expanding the framework to accommodate external calls could introduce new opportunities to integrate with centralized platforms and external services, thus improving the flexibility and attractiveness of holochain as a platform for decentralized application development. Although delving into such enhancements is beyond the scope of this thesis, it presents a promising area for future exploration. Examining the feasibility, consequences, and technical approaches for enabling external communications within the holochain framework could significantly advance decentralized technologies and their integration with the current digital infrastructure.

7.4 Evaluation

The evaluation of the websocket client applications within and outside the holochain framework reveals critical insights into the framework's architectural and operational constraints, particularly concerning external network communications. Our investigation began with the premise that distinct operational environments (within and outside the holochain framework) would yield different outcomes in terms of the applications' ability to establish connections and exchange data. The effective functioning of the websocket clients in settings outside of holochain illustrated their strength and adherence to common web communication standards. However, the noticeable difference in their performance when used within the holochain framework, notably the challenge of connecting to the centralized platform on Amazon Web Services, underscored notable constraints in the framework's facilitation of external network interactions, along with its shortcomings experimentally. This disparity underscores a core element of the holochain structure that emphasizes internal network security and reliability at the expense of external linkages, consequently restricting its ability to seamlessly integrate with diverse platforms and services.

Further examination revealed that holochain intentionally restricts the ability to make external calls to maintain its decentralized integrity, which poses a significant challenge for applications requiring interactions with external centralized platforms. This architectural decision underscores a philosophical and practical divergence between decentralized applications (DApps) developed within the holochain ecosystem and traditional centralized applications. The inability to establish integrative connections with external agents or platforms such as AWS from within the holochain framework not only limits the scope of potential applications, but also raises questions about the framework's adaptability and scalability in real-world scenarios. Although these limitations are in place to protect the decentralized nature of applications, they also underscore the need for a balance between security, decentralization, and interoperability in the development of DApps. This could present an issue for indigenous communities desiring some level of integration with contemporary platforms, or on-site computing if they need computational power as a partial investment towards achieving self-sustainability.

Considering the open-source nature of the holochain project, there lies a potential for future modifications to the framework that could enable external calls and, by extension, broaden the

scope of its applicability and utility. Such modifications would require a thoughtful redesign of the framework's architecture to incorporate external connectivity without compromising its core principles of decentralization and security. This area of development presents fertile ground for future research, potentially paving the way for a new generation of DApps that can seamlessly integrate with external platforms and services. However, it is important to note that any modifications to enable external connections must be approached with caution to ensure that the fundamental advantages of the holochain framework, such as data integrity, user autonomy, and network resilience, are not undermined. This evaluation underscores the complex interplay between architectural decisions, application requirements, and the overarching goals of decentralized systems, highlighting the need for ongoing research and development in this field. Currently, efforts [129] are underway to develop a platform that will allow holochain applications to communicate with external sources outside the holochain network.

Chapter 8

Discussion

Although holochain provides standard functionalities, its effective implementation and maintenance require a certain level of technical proficiency. Indigenous communities may need assistance and resources to navigate the technical intricacies of the system. The implementation and maintenance of a holochain-based system require various resources, such as hardware, experience in software development, and continuous support. Limited resources can present difficulties, particularly for smaller or underfunded indigenous communities. This discussion emphasizes the importance of adopting a comprehensive approach that integrates technical innovation with ethical considerations to navigate the intricate realm of indigenous data sovereignty in the digital era.

8.1 Research Questions

8.1.1 Can holochain technology be used to support indigenous data sovereignty?

Drawing from our findings in the second and third phases of our research, where we explore in depth the concept of holochain, the process of creating a holochain application, possible integrations, client-side development, and the challenges faced during the construction of our prototype, we find that holochain emerges as a promising technology for supporting indigenous data sovereignty in spite of its limitations and disadvantages. These limitations and drawbacks are discussed in subsequent sub-sections. The clearly noticeable features that we have identified are as follows.

- Holochain is implemented using Rust [124]. As discussed in earlier sections, Rust is a modern systems programming language that ensures memory safety. It is employed in zero trust architectures due to its stringent borrow checker, which helps to maintain integrity by preventing errors in ownership and managing low-level resources and immutability processes [123]. This foundational fact is important for indigenous data sovereignty to ensure data integrity and data ownership at a fundamental level isn't open to exploitation through simple means despite the fact that this is dependent on holochain's underlying architecture, the validation rules that may

be implemented for stringent data checking is employable within the framework as outlined in their documentation [145], [157].

- Holochain’s mutual sovereignty. Considering that one of the core principles of indigenous data sovereignty is data ownership [57], holochain implements mutual sovereignty¹, meaning that every participant on the network shares equal responsibilities and authority. This ensures that network participants can maintain privacy over their individual data while managing public data using shared compute resources and storage through DHT sharding [147]. In our phase two implementation of the prototype V2, we did not implement this feature of ensuring private entries as this did not align with what we wanted to achieve and build upon, instead we focused on ensuring that data was shared and that the feature was observable to other agents who would be concerned over such data. Although we did not display private entries as we could have, we were able to verify through experimentation that the feature was available.
- Holochain’s road map and trajectory. Considering the current progress of holochain and its path forward, including the introduction of extensive testing and pub sub features [146]. Holochain is evolving to become a potentially valuable technology not only for indigenous data sovereignty but also for various socio-economic challenges and daily applications. Currently, holochain is already applied in other fields like energy grids for peer-to-peer microtransactions, among others [148].

8.1.2 What are the key requirements for indigenous data sovereignty and does holochain meet those requirements?

The primary criteria we concentrated on regarding indigenous data sovereignty were indigenous authority and control over the form of the data within an application. This included the incorporation of indigenous worldviews in the design of the application, such as culture, naming conventions, and data ownership [31], [57]. In the second phase of our research and while examining the present literature on holochain, we were persuaded that holochain could fulfill key requirements for indigenous communities. This is due to holochain’s framework offering mutual accountability and a privacy-first model based on both its existing and prospective features. Additional supporting features include encrypted data during transit between peers within the same network, ensuring that no unauthorized agent can join their respective network [149]. Although we did not showcase the mentioned encryption features, these are evident in holochain’s code base [144]. We incorporated cultural elements such as Pepeha [154] into our application design and user interface components 6.1, showcasing how members of the indigenous community would prefer to view their data and how these data entries should be managed within the network. In addition, we crafted basic validation rules that could serve as examples of valid entries, tailored to indigenous communities. These indigenous communities could create their

¹Though this concept is not unique to holochain.

own validation rules to be housed within their networks, as the rules and their complexities are shaped by the stakeholders' preferences and needs.

- Integrity types. Within our application, we succeeded in sampling data models important to us by use case based on geographic regions within New Zealand 6.2. This sample alone demonstrated our ability to interpret data in ways significant to indigenous communities. An example of this would be the correlation between english names of places and their corresponding indigenous names being modeled within the application. This approach enables data interpretation to remain within the boundaries of the community, with the data stored on participants' devices through the sharding of the distributed hash table. As a result, all participants would essentially become representatives of their community, even if they are spread out in different locations.
- Interface design. Since the holochain application and peer entry into the network can be accomplished through various means, indigenous communities have the liberty to showcase significant artifacts, themes, and histories pertinent to their heritage. This supports both group identity and individual identity rooted in cultural importance. We demonstrate our small sample of this here 5.11

8.1.3 What are the potential benefits and challenges experienced when adopting holochain for indigenous data sovereignty?

As we progressed through phases involving cloud infrastructure setup, holochain application development, and integration efforts, we became convinced of the substantial benefits offered by holochain technology. This has been affirmed through earlier discussions. The primary advantage is ownership of data. Holochain embeds data ownership and management as core principles, providing resource efficiency since data does not need to traverse every node but its neighbors [145], unlike in traditional blockchains where all validators must process the block of transactions to validate it [79]. Although the benefits are clear, we find it important to note the challenges for indigenous sovereignty.

- Operating system limitations. It was evident when building a holochain application, we were required to have at minimum the recommended specifications by holochain in order to test and build the applications [141]. However, these recommendations aren't limited to developers, but also to consumers of this holochain. Currently, holochain is limited to desktop operating systems like Windows and Linux, but there is no official support for web clients and mobile devices which we believe is critical to modern utilization of everyday tasks. However, given that there are clients for Rust and even C# which is the community maintained repository, there are possibilities for holochain to be extended into mobile devices but we have yet to see a holochain application produced on these devices.
- Integration. While there may be initial perceived challenges with integrating holochain, such integration could be advantageous for indigenous communities seeking to optimize data processing.

An on-premise solution could facilitate the processing of data imports into a community-based holochain network. Nonetheless, this approach presents its own difficulties, including the collaborative efforts required [16] between developers from indigenous communities and core holochain contributors regarding dependencies.

- **Limited technical resource.** When developing holochain applications, indigenous communities will need technical expertise with the programming language Rust. Despite Rust's growing popularity in recent years [131], there is a global shortage of proficient developers, probably due to its challenging learning curve. It may take substantial effort to recruit developers to work on projects that use Rust, especially when aligned with the specific needs of indigenous communities. Our own experience learning Rust and the holochain framework has highlighted the significant effort required from our work in phase two.
- **Limited compute resource.** In the initial phase of our experiment, we utilized Amazon's centralized platform instead of an on-premise solution to illustrate a hybrid approach as a level of IDS with an open source technology like holochain. This method contradicts the standpoint of indigenous data sovereignty proponents, who oppose the use of large data companies [126], particularly in today's era where AI-driven data usage raises privacy concerns and potentially unpredictable biased outcomes against marginalized communities [135]. Our aim was not to contest the views of indigenous data sovereignty advocates but to demonstrate through experimentation a hybrid approach in a ready-to-deploy production environment that could theoretically be replaced and replicated by an on-premise solution with adequate compute resources, albeit we lacked the time and financial means to do so. This underscores that communities wishing to use modern software like holochain must indeed have appropriate hardware resources available, particularly if they choose a hybrid approach with modern practices and the use of existing capable tooling.

8.2 Limitations

We present our results with some constraints and discuss their applicability under various factors. Firstly, holochain applications require internet connectivity to participate in the sharing of public data, which cannot be guaranteed for all indigenous communities and was not tested experimentally since our experiment was conducted in a controlled environment with substantial internet availability, without any means of testing in rural areas. Although holochain supports peer-to-peer communication and consensus, this dependence on the internet poses challenges for indigenous communities that may have limited or unreliable access based on their locations. Secondly, as previously discussed, the primary programming language for holochain, Rust, could pose a challenge for indigenous communities that might lack the technical abilities or knowledge required to create and manage applications on this platform. Furthermore, holochain's current ecosystem and user base are still relatively small compared to more established blockchain platforms. Therefore, there may be a lack of resources, tools, and

support specifically tailored to indigenous communities' needs when it comes to implementing and utilizing holochain for data sovereignty purposes. Furthermore, these difficulties were evident during our experiments, ranging from the unstable connection formation of our websocket clients 7.3, to our attempt in developing a holochain application before noticing that holochain entries are by default public.

As indicated in 8.1.3, the final point concerning limited computing resources, our use of a cloud provider was intended to illustrate a form of IDS using open-source software. This constraint in our experiment might have posed different challenges that could have been addressed had we had sufficient computing and financial resources. Consequently, we acknowledge that our investigation, while offering a reproducible experiment with a cloud provider, does not necessarily yield a replicable solution for communities with access to on-premise computing resources.

Our findings cannot be generalized to all indigenous communities due to varying levels of resources among communities with different characteristics and accessibility. This underscores the point that technological advancements suitable for one community may not be pertinent to another where the importance of food resources takes precedence over technological solutions for information preservation and retention. Nevertheless, our results are limited to communities with adequate resources and concern for cultural preservation and information retention, as well as the risk of misuse of their identity and representation. In addition, tools that have been refined and advanced by holochain, like the holochain scaffolding tool, can significantly cut down the development time for developers who aim to provide technological solutions to communities with such needs. Furthermore, the outcomes of our experiment do not consider the dynamics of teams with more than two developers working on this type of project, which could result in different findings even under the same time limitations. Another limitation to acknowledge is that our study does not consider the subtle intricacies of specific indigenous modules or designs that support the depiction and verification of indigenous oral traditions, traditional ecological wisdom, and customary laws within holochain application data frameworks. Working together with indigenous elders, knowledge custodians, and cultural consultants will be essential for creating these modules in a manner that respects the authenticity and sanctity of indigenous knowledge.

In conclusion, despite our discussion of time limitations, indigenous communities may not face the same constraints we do, either positively or negatively. We encountered a significant challenge related to time constraints during each phase of our research, further exacerbated by the absence of effective stakeholder management. The extensive scope of this thesis, coupled with the substantial effort required to immerse ourselves in the subject matter, made it unfeasible to meaningfully involve stakeholders. This situation led to considerable stress for our team, which impeded the smooth progress of our experiments as initially anticipated. These challenges not only caused delays in our research schedule, but also turned what should have been an exciting exploration into a source of frustration. Upon reflection, it is evident that the convergence of emerging technological paradigms and academic research poses distinct challenges that necessitate careful navigation.

8.3 Critical Analysis

Throughout our investigation and subsequent phases of project implementation, we collectively gained a deep understanding of the complexities associated with possible integration strategies with holochain. This understanding was a result of our thorough analysis, which revealed that despite its innovative potential, holochain is still in its early developmental stages. Consequently, we recognized that promoting its use in large-scale production systems as a feasible solution for significant challenges may be premature at this stage. Our assessment emphasized that while holochain presents a fresh approach to decentralized applications, its technological maturity and ecosystem development are not yet at a level that would encourage widespread adoption by developers worldwide. This realization was a critical point in our study, highlighting the disparity between the theoretical promise of holochain and its practical implementation in the current technological environment. We encountered significant challenges due to the steep learning curve associated with using holochain sufficiently, despite leveraging online documentation and conducting numerous experiments to comprehend its foundational concepts. It became increasingly apparent to us that the time investment required to become proficient in holochain technology substantially impacted our progress. Even after we had gained a fundamental understanding of its principles, we realized that additional time was necessary to develop software solutions that could be used effectively by the broader community. This revelation was a critical insight for us and highlighted the complexities of working with emerging technologies.

The usability of holochain is one aspect that stands out, particularly its lack of clear conceptual documentation bridging theory and practice. Although there is a scaffolding tool available, it does not significantly enhance my understanding of the topic; rather, it only shows that I can create an application in holochain. There appears to be a gap between the scaffolding tool, the theoretical concepts, and the provided documentation. These noticeable discrepancies underscore the need for a more cohesive integration of different layers, allowing developers and researchers to confidently grasp the concepts and build applications by synthesizing their knowledge from a centralized paradigm to a decentralized one. Furthermore, our investigation, paired with indigenous data sovereignty with a focus on māori sovereignty principles, revealed how significant it was to include stakeholders of indigenous knowledge to further our prototype productively. When we juxtaposed these principles with the current status of holochain technology and the wider industry landscape, we realized that the scope of implementation would require some significant effort financially and in time relative to the project proposal. Considering that there are already established technologies on the market, such as blockchain, which have the advantage of a larger developer community and more significant financial backing, other technologies appear to be better suited to address these critical issues. This comparative analysis reveals challenges that are not only technical but also financial and socioeconomic.

An additional point is that holochain restricts its framework's flexibility by not including more front-end client-facing frameworks and lacking the capacity to integrate with external service providers within its *backend*. In our view, this connectivity deficit limits the alternatives available to indigenous communities seeking to investigate various methods for managing, positioning, or collecting their data. Indigenous data sovereignty, along with the CARE principles which emphasize the authority

to control their data [57], should incorporate flexibility within the holochain framework as a built-in feature, rather than simply asserting its open-source nature as sufficient. However, we believe that this flexibility should be a supported feature, complete with documentation to foster socioeconomic benefits for communities seeking options with reasonable constraints. In conclusion, despite these drawbacks associated with holochain, it holds the potential to extend its reach beyond its current roadmap to areas beneficial to indigenous communities and their unique technological challenges.

8.4 Possibilities for Further Research

The incorporation of holochain applications into indigenous communities presents an opportunity for technological empowerment and advancement. However, this process is not without challenges. It requires a comprehensive approach that takes into account not only technological considerations but also cultural sensitivity, ethical principles, and social fairness. The success of this endeavor should not be solely judged based on the implementation of technology, but rather on its ability to improve the self-determination and welfare of indigenous communities. It should respect indigenous values, safeguard indigenous rights, and honor indigenous cultural heritage. More research is necessary to explore and address the following areas and the impact of holochain on traditional governance systems in indigenous communities and the potential to improve local decision-making processes.

To strengthen holochain's contribution to indigenous data sovereignty as highlighted in our study, we recommend the following measures to empower indigenous communities to achieve self-sustainability with these technologies without significantly sacrificing performance. This call to action, we believe, would help in the effective implementation of holochain in indigenous contexts. These recommendations should cover technical, social, and cultural aspects to ensure that holochain adheres to the principles of indigenous data sovereignty and respects the specific requirements of indigenous communities.

- Opt-in network selection for reliability of addressing communities in rural area's that have very limited access to internet activity and still wish to continue sharing data amongst their closes neighboring nodes. This would enable rural communities to have the same benefits as those with reliable network over the internet.
- The integration of indigenous knowledge systems with technological advancements within the holochain ecosystem and its influence on cultural conservation and revitalization. An example of enhancement could be the deployment of localized artificial intelligence, either on-device or on-premise, managed by indigenous communities and integrated with specific holochain networks. Local language models tailored for indigenous communities and their cultural values can offer deeper understanding of important data points for these communities.
- Mobile applications designed for managing medical records and data sovereignty for indigenous communities would be advantageous, as individuals in modern times have handheld devices and

appear to be more efficient with information on portable devices [128]. The widespread use of mobile technology, compared to computers, would aid in the global practical implementation of holochain.

- Develop frameworks for decentralized technologies designed for indigenous communities to evaluate the social, cultural and economic impacts of solutions like holochain. These frameworks should include assessments of appropriation and self-determination, taking into account feature disparities and other important factors relevant to indigenous communities.

Future research can make valuable contributions to the ongoing enhancement and fine-tuning of the integration of holochain in indigenous settings, thereby advancing the principles of indigenous data sovereignty and self-determination. The integration of holochain into indigenous contexts has the potential to significantly empower and enable indigenous communities to exercise self-determination. Taking into account the recommendations mentioned above, holochain can effectively bridge the gap between technological innovation and the preservation of indigenous knowledge and governance systems. In conclusion, a successful integration of holochain in indigenous contexts requires a thoughtful blend of technological progress, cultural recognition, and community empowerment. By embracing these recommendations, holochain can effectively facilitate indigenous data control while upholding the principles of self-governance, consent, and community-driven administration.

8.5 Conclusion

After a thorough evaluation of the potential benefits and challenges associated with integrating holochain in indigenous contexts, it is evident that a thoughtful and all-encompassing approach is crucial to uphold the principles of indigenous data sovereignty. A detailed examination has underscored the importance of tackling issues concerning infrastructure and connectivity, power imbalances, data security, cultural preservation, and skills enhancement. These recommendations are essential for the successful implementation of holochain in indigenous societies. By adopting a holistic strategy that includes support for infrastructure and connectivity, collaborative design and empowerment tactics, stringent security and confidentiality measures, cultural assimilation and acknowledgment initiatives, as well as capacity development and educational programs, it is possible to tackle the identified obstacles and create a more inclusive and empowering space for indigenous communities within the holochain framework.

In conclusion, our research results underscore the significance of holochain in the realms of indigenous data sovereignty and software engineering through data analysis and experimentation. Future research in this area has the potential to drive substantial progress in software engineering and indigenous data sovereignty. This research not only enriches the existing knowledge base, but also sets the stage for future investigations. Essentially, our study acts as a foundation for enhancing the understanding of leveraging decentralized technologies such as holochain to support indigenous

data sovereignty within cultural communities, stressing the importance of ongoing exploration and creativity in this field.

Appendix A

Symbols Used In The Thesis

In this chapter, we provide a summary of the symbols and abbreviations utilized throughout this thesis. Although the items are not listed in a specific sequence, we have made an effort to arrange them by their relevant context. Further properties, particularly applicable in specific contexts, are delineated as required in the main chapters of this thesis.

A.1 Distributed Systems Symbols

Name	Symbol	Description
Distributed System	Ω	A distributed system
Node	n	A node within the distributed system Ω
Node State	σ_n	The state of a node n .
Hash Function	H	A cryptographically secure hash function
Validation Function	$V(t, v)$	A <i>validation</i> function with t being some input along with additional validation data v
Channel	C	The channel that allows for nodes N to communicate to one another
Data Elements	D	The non-hashed chain <i>data elements</i> within the distributed system Ω
A Machine	M	Some virtual or non-virtual machine.
Neighboring Size	q	Represents the size of the neighboring nodes in N surrounding a node n in a system Ω
Time Complexity	$c + \lceil \log(m) \rceil$	The time complexity derived [1] to approximate a DHT lookup.
Entry Validation	$v(n . m)$	The application-specific complexity and validation routines that are unknown and are assumed to represent the overall complexity of a system.

Table A.1: Distributed Systems Symbols

A.2 Holochain Distributed Systems Symbols

Name	Symbol	Description
Holochain Distributed System	Ω_{hc}	A holochain distributed system
Node	n	A node within the distributed system Ω
Set of Participants	N	Represents the set of elements $\{n_1, n_2, n_3, \dots, n_n\}$ or participants in the holochain system. These elements are commonly known as nodes or agents
Node State	σ_n	The state of a node n .
Set of States	S_n	Represents the set of states $\{\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_n\}$ of a node n
Hash Function	H	A cryptographically secure hash function
Hash Chain / Initial Entry	X_i / X_n	Indicative of the hash-chain within the distributed system Ω . Within a holochain system Ω_{hc} this is indicative of the initial entry X_n
Second Entry	t_n	The second entry t_n of all initial entries X_n and be a set of $\{p, i\}$ where p is the public key and i is the identifying information relative to the agent / node.
DNA	DNA	The set of elements $\{e_x, \dots, f_x, \dots, p_x, \dots\}$ containing entry types e_x that can be added to the chain, functions f_x executable on a machine M and various properties of the system p_x relative to M
Functional executable machine	F_{app}	Machine capable of running executable code such that $F_{app} = \{app_1, app_2, \dots, app_n\}$.
Validation function	F_v	The validation function in respect to app_n of F_{app}
System entry validation	$V_{sys}(e_x, e, v)$	The entry specificity where e is the form required, e_x the entry definition and v indicative of the output.
Exposed function	F_I	An exposed function that is a subset of F_{app} with an available set of actions.
DHT	DHT_{hc}	A validating, sharded distributed hash table within a distributed system that is holochain Ω_{hc}
DHT State	Δ	The distributed hash table state particularly within a holochain system Ω_{hc}
Neighboring Size	q	Represents the size of the neighboring nodes in N surrounding a node n in a holochain system Ω_{hc}

Table A.2: Holochain Model Attributes [145]

Name	Symbol	Description
DHT functions	dht_n	The set of functions that operate on the DHT $F_{DHT} = \{dht_{get}, dht_{put}, \dots, dht_n\}$
Key values	σ_n	The key values $\sigma_{key,value}$ or $\sigma(k, v)$ in the holochain DHT state Δ_{hc} .
System functions	sys_n	The set of system functions $F_{sys} = \{sys_{commit}, sys_{get}, \dots, sys_n\}$ accessible given a set of operational conditions.
Private entry	$\sigma_{private}$	A private entry.
Neighboring nodes	n_n	Nodes in the neighborhood around a particular node n .
Redundancy factor	r	The redundancy factor parameter of a DHT_{hc} characteristic.

Table A.3: Holochain Model Attributes [145]

Appendix B

Amazon Web Services

Amazon Web Services is one of the leading providers of cloud computing solutions that can be utilized to replicate the modern application architecture explored in this dissertation. Due to its adaptable on-demand capabilities, AWS has become a crucial part of the IT infrastructure landscape. It offers a wide variety of services, including APIs and platforms, that meet the needs of individuals, companies, and government agencies. AWS provides features like data storage, computing power, and network resources, allowing users to quickly and securely create and deploy applications.

AWS Service	Category	Outline
API Gateway	API	REST / Websockets / Protocol
Cloudwatch	Logging	Security audits
Cognito	Authentication and Authorization	Security
DynamoDB	NoSQL Database	Scalable unstructured data

Table B.1: AWS Services - Overview

B.1 AWS API Gateway

AWS API Gateway is a completely managed service that simplifies the process for developers to generate, release, and oversee APIs of any size. By utilizing the API Gateway, developers can construct secure and expandable APIs to facilitate communication between various parts of their applications or to make their services accessible to outside clients.[64].

B.2 AWS Cloudwatch

CloudWatch gathers and retains metrics, logs, and events from both AWS services and custom applications. It delivers immediate monitoring and alerting functions, empowering users to promptly

detect and address any concerns. AWS CloudWatch is a monitoring and observability service compliant with AWS standards, furnishing essential insights into the performance and status of applications, infrastructure, and services operating on the AWS platform. It accumulates and preserves metrics, logs, and events from AWS services and custom applications, offering real-time monitoring and alerting features to assist users in promptly recognizing and resolving any issues [64].

B.3 AWS Cognito

AWS Cognito is a platform that offers user authentication, authorization, and user administration for web and mobile applications ¹. It allows you to efficiently incorporate sign-up and sign-in functionalities into your applications, securely store application data in the AWS Cloud, verify users via social identity providers like Facebook or Google, and grant temporary access credentials to AWS services from your application backend with precise permissions using IAM roles.[64].

B.4 AWS DynamoDB

Amazon DynamoDB is a NoSQL database service that is fully managed, providing fast and consistent performance along with easy scalability. It allows developers to store and retrieve data quickly and efficiently, ensuring low latency and high throughput.[64].

B.5 AWS Lambda

Lambda from Amazon Web Services is a robust serverless computing service that empowers developers to execute code without the need to handle servers, thereby enhancing scalability and cost-effectiveness. In this study, AWS Lambda will be utilized to trigger functions in the messaging and streaming layer of the application architecture [64]. By leveraging Lambda functions, developers can integrate real-time analysis and processing of streaming data into their DynamoDB tables. This is accomplished by configuring event triggers that activate Lambda functions whenever data is inserted or modified in the tables. By incorporating DynamoDB Accelerator, which offers a highly accessible in-memory cache, the real-time analysis and processing of streaming data can be significantly expedited, resulting in a performance enhancement of 10 times, reducing the processing time from milliseconds to microseconds [82].

B.6 AWS RDS

Amazon RDS, a database service managed by AWS, simplifies database management by providing SQL-like functionality, supporting a flexible document data model, and offering robust transactional

¹<https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html>

capabilities. It eliminates the need for manual server management and enables effortless scalability to accommodate application requirements. Cost-effectiveness is a key feature of Amazon RDS, as users only pay for the resources they consume. Additionally, Amazon RDS supports various relational database engines, such as MySQL, PostgreSQL, Oracle, and SQL Server, which can be seamlessly integrated with other AWS services and scaled to handle fluctuating workloads [64].

Database Type	Use Case	AWS Service
Relational	Traditional, ERP, CRM	Aurora, RDS, Redshift
Key-value	High-traffic, e-Commerce	DynamoDB
In-memory	Caching, session, geospatial	Elasticache, MemoryDB
Document	Catalogs, user profiles, content	DocumentDb

Table B.2: **AWS Service** - Database type

As seen in B.2, we will choose a relational database handled by the Aurora service for serverless support.

B.7 AWS S3

Amazon Simple Storage Service (S3), offered by AWS, is a dependable and scalable service for storing objects. It is a cost-effective solution for securely retaining large volumes of unstructured or partially structured data. Amazon S3 can function as a highly accessible key-value repository in a serverless web application or site [64]. Organizations seeking to store and oversee their data in the cloud need to take into account AWS adherence [36]. AWS has implemented robust measures for security and data safeguarding in the cloud, which are grounded on the five pillars of AWS: security, reliability, performance efficiency, cost optimization, and operational excellence. The AWS whitepaper delineates the primary features and benefits of AWS security and adherence, such as.

- **Privileged user access controls:** Organizations can use AWS to set up and manage user access rights and permissions to protect confidential data from unauthorized access.
- **Data encryption:** AWS offers a range of encryption methods to ensure that data are safeguarded while in transit and when stored.
- **Data Location and Segregation:** provides organizations with the option to select the region in which their data is stored, thus ensuring that they meet data sovereignty and regulatory requirements.
- **Data Backup and Recovery:** Amazon Web Services provides integrated data backup and restoration solutions to guarantee data availability and accuracy.

- **Physical and environmental controls:** Amazon Web Services data centers are equipped with the latest security measures, including access control systems, surveillance cameras, and fire suppression systems, to protect physical infrastructure and prevent unauthorized access.
- **Compliance with industry standards:** Amazon Web Services meets a variety of industry standards and certifications, such as SOC 1/ISAE 3402, SOC 2, SOC 3, FISMA, HIPAA, GDPR, and PCI-DSS.
- **Security incident response:** Amazon Web Services has implemented systems and procedures to quickly respond to and reduce the impact of security incidents, guaranteeing ongoing protection of customer information. These features and benefits of AWS security and compliance make it a reliable option for companies looking to store and manage their data in the cloud.

Besides the security and compliance functionalities mentioned previously, AWS also offers extensive monitoring and logging features. These enable entities to observe and assess their system operations, providing essential insights into possible risks and weaknesses. As an example, AWS CloudTrail documents all API requests made in an AWS account, generating a thorough log of actions that can support audits and forensic examinations.[58], [61].

AWS operates on a shared responsibility model, where they oversee the security of the cloud infrastructure, while customers are accountable for safeguarding their applications and data within the cloud [64], [87]. To assist customers in fulfilling their security responsibilities, AWS presents a range of tools and resources. For instance, AWS Identity and Access Management permits organizations to regulate user access and permissions, and AWS CloudFormation streamlines the secure and controlled deployment and management of resources [36]. This collaborative model guarantees that both AWS and its customers collaborate to uphold a high level of security for cloud-stored data [58]. AWS implements robust security measures and compliance certifications to ensure the safeguarding of customer data.

References

- [1] P. Maymounkov and D. Mazières, “Kademlia: A peer-to-peer information system based on the xor metric,” in *Peer-to-Peer Systems*, P. Druschel, F. Kaashoek, and A. Rowstron, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 53–65, ISBN: 978-3-540-45748-0.
- [2] B. Cohen, “Incentives build robustness in bit-torrent,” 2003.
- [3] A. Hevner, S. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS Q.*, 2004. DOI: 10.2307/25148625.
- [4] M. Koubarakis, “/ subscribe systems with distributed hash tables and languages from ir [extended,” 2004.
- [5] D. Qiu and R. Srikant, “Modeling and performance analysis of bittorrent-like peer-to-peer networks,” *SIGCOMM '04*, 2004. DOI: 10.1145/1015467.1015508.
- [6] O. Babaoglu, M. Jelasity, A. Montresor, C. Fetzer, and S. Leonardi, Eds., *Self-star Properties in Complex Information Systems: conceptual and Practical Foundations*. Berlin, Heidelberg: Springer-Verlag, 2005, ISBN: 3540260099.
- [7] M. Castro and R. v. Renesse, “Peer-to-peer systems iv,” *Lecture Notes in Computer Science*, 2005. DOI: 10.1007/11558989.
- [8] B.-G. Chun, B. Y. Zhao, and J. Kubiatowicz, “Impact of neighbor selection on performance and resilience of structured p2p networks,” *Lecture Notes in Computer Science*, 2005. DOI: 10.1007/11558989_24.
- [9] “Centralization, organizational strategy, and public service performance,” *Journal of Public Administration Research and Theory*, vol. 19, no. 1, pp. 57–80, 2007. DOI: 10.1093/JOPART/MUM039.
- [10] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” *J. Manag. Inf. Syst.*, 2007. DOI: 10.2753/MIS0742-1222240302.
- [11] L. Jian and J. K. MacKie-Mason, “Why share in peer-to-peer networks?” *ICEC*, 2008. DOI: 10.1145/1409540.1409546.

- [12] “A strategic central approach to data collection and integration: A case of a research- intensive university,” vol. 15, no. 1, pp. 1–8, 2010.
- [13] *Authorized file-sharing system on P2P networks*, ACM, 2010, pp. 1289–1293. DOI: 10.1145/1815396.1815692.
- [14] *Centralized Management of Data Collection over Hybrid Networks*, IEEE, 2010, pp. 180–185. DOI: 10.1109/INTERNET.2010.39.
- [15] “A study on the group routing algorithm in dht-based peer-to-peer system,” *Journal of the Korea Society of Computer and Information*, vol. 17, no. 12, pp. 111–120, 2012. DOI: 10.9708/JKSCI/2012.17.12.111.
- [16] A. D. Bakar, Y. H. Sheikh, and B. Sultan, “Opportunities and challenges of open source software integration in developing countries: Case of zanzibar health sector,” *Journal of Health Informatics in Developing Countries*, 2012.
- [17] S. Gregor and A. Hevner, “Positioning and presenting design science research for maximum impact,” *MIS Q.*, 2013. DOI: 10.25300/MISQ/2013/37.2.01.
- [18] A. Stefik and S. Siebert, “An empirical investigation into programming language syntax,” *TOCE*, 2013. DOI: 10.1145/2534973.
- [19] *Design Science Methodology for Information Systems and Software Engineering*. 2014.
- [20] H. K. Lu, *Keeping your api keys in a safe*, 2014. DOI: 10.1109/CLOUD.2014.143.
- [21] “Re-chord: A self-stabilizing chord overlay network,” *Theory of Computing Systems Mathematical Systems Theory*, vol. 55, no. 3, pp. 591–612, 2014. DOI: 10.1007/s00224-012-9431-2.
- [22] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. Kwak, “The internet of things for health care: A comprehensive survey,” *IEEE Access*, 2015. DOI: 10.1109/ACCESS.2015.2437951.
- [23] F. E. Knowles and L. L. Lovern, “The united nations declaration on the rights of indigenous peoples,” 2015. DOI: 10.1057/9781137557452_7.
- [24] T. Kukutai and M. Walter, “Recognition and indigenizing official statistics: Reflections from aotearoa new zealand and australia,” *Statistical journal of the IAOS*, 2015. DOI: 10.3233/SJI-150896.
- [25] A. Brock, *Ceptr Under the Hood: The full overview of Ceptr*, May 2016. [Online]. Available: <http://ceptr.org/blog/2016-05-04-ceptr-under-the-hood>.
- [26] “Centralization and the success of erp implementation,” *Journal of Enterprise Information Management*, vol. 29, no. 5, pp. 728–750, 2016. DOI: 10.1108/JEIM-07-2015-0058.
- [27] M. Davis, “Data and the united nations declaration on the rights of indigenous peoples,” 2016. DOI: 10.22459/CAEPR38.11.2016.02.

- [28] R. Jansen, “Indigenous data sovereignty: A māori health perspective,” 2016. DOI: 10.22459/CAEPR38.11.2016.11.
- [29] A. Khan, G. Segovia, and D. Kossmann, “On smart query routing: For distributed graph querying with decoupled storage,” *USENIX Annual Technical Conference*, 2016.
- [30] T. Kukutai and J. Taylor, “Data sovereignty for indigenous peoples: Current practice and future needs,” 2016. DOI: 10.22459/CAEPR38.11.2016.01.
- [31] T. Kukutai and J. Taylor, “Indigenous data sovereignty: Toward an agenda,” 2016.
- [32] *Mutual Credit Cryptocurrencies: Beyond blockchain bottlenecks*, Mar. 2016. [Online]. Available: <http://ceptr.org/whitepapers/mutual-credit>.
- [33] C. Snipp, “What does data sovereignty imply: What does it look like?,” 2016. DOI: 10.22459/CAEPR38.11.2016.03.
- [34] J. R. Venable, J. Pries-Heje, and R. L. Baskerville, “Feds: A framework for evaluation in design science research,” *European Journal of Information Systems*, 2016. DOI: 10.1057/EJIS.2014.36.
- [35] S. Mastorakis, A. Afanasyev, Y. Yu, and L. Zhang, “Ntorrent: Peer-to-peer file sharing in named data networking,” *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, 2017. DOI: 10.1109/ICCCN.2017.8038462.
- [36] M. Stigler, “Amazon web services,” 2017. DOI: 10.1007/978-1-4842-3084-8_3.
- [37] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An overview of blockchain technology: Architecture, consensus, and future trends,” *2017 IEEE International Congress on Big Data (BigData Congress)*, 2017. DOI: 10.1109/BIGDATAACONGRESS.2017.85.
- [38] J. Atkinson, “Privacy, ethics and information sharing of new zealand’s integrated data infrastructure (idi): A discussion of the issues, challenges and opportunities,” *International Journal for Population Data Science*, 2018. DOI: 10.23889/IJPDS.V3I4.689.
- [39] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, “Healthcare blockchain system using smart contracts for secure automated remote patient monitoring,” *Journal of Medical Systems*, 2018. DOI: 10.1007/s10916-018-0982-x.
- [40] I.-S. Hwang, A. Rianto, and A. F. Pakpahan, “Software-defined peer-to-peer file sharing architecture for twdm pon,” in *2018 27th Wireless and Optical Communication Conference (WOCC)*, 2018, pp. 1–4. DOI: 10.1109/WOCC.2018.8372713.
- [41] Z. G. Ziqian Meng Zhong Chen, “Re-design the bittorrent protocol in next generation expressive internet architecture,” in *Proceedings of 2018 the 8th International Workshop on Computer Science and Engineering*, 2018, pp. 476–482. DOI: 10.18178/wcse.2018.06.082.
- [42] “A novel massive deployment solution based on the peer-to-peer protocol,” *Applied Sciences*, vol. 9, no. 2, pp. 296–, 2019. DOI: 10.3390/APP9020296.

- [43] A. Agarwal and S. Kamara, "Encrypted distributed hash tables," *IACR Cryptology ePrint Archive*, 2019.
- [44] A. Agarwal and S. Kamara, "Encrypted distributed dictionaries," *Cryptology ePrint Archive, Paper 2019/1126*, 2019, <https://eprint.iacr.org/2019/1126>. [Online]. Available: <https://eprint.iacr.org/2019/1126>.
- [45] C. C. Agbo, Q. Mahmoud, and J. M. Eklund, "Blockchain technology in healthcare: A systematic review," *Healthcare*, 2019. DOI: 10.3390/HEALTHCARE7020056.
- [46] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telematics and Informatics*, 2019. DOI: 10.1016/J.TELE.2018.11.006.
- [47] S. Chen, K. R. Choo, X. Fu, W. Lou, and A. Mohaisen, Eds., *Security and Privacy in Communication Networks - 15th EAI International Conference, SecureComm 2019, Orlando, FL, USA, October 23-25, 2019, Proceedings, Part II*, vol. 305, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer, 2019, ISBN: 978-3-030-37230-9. DOI: 10.1007/978-3-030-37231-6. [Online]. Available: <https://doi.org/10.1007/978-3-030-37231-6>.
- [48] *Design Science Research for Development and Validation of a Pedagogical Agent for STEM Education*, 2019. DOI: 10.1109/LACLO49268.2019.00031.
- [49] T. Kukutai and D. Cormack, "Mana motuhake ā-raraunga: Datafication and social science research in aotearoa," *Kotuitui: New Zealand Journal of Social Sciences Online*, 2019. DOI: 10.1080/1177083X.2019.1648304.
- [50] R. Lovett, V. Lee, T. Kukutai, D. Cormack, S. C. Rainie, and J. D. Walker, "Good data practices for indigenous data sovereignty and governance," 2019.
- [51] Z. Meng, Z. Chen, and Z. Guan, "Peer-to-peer file sharing in next generation expressive internet architecture," *CCF Transactions on Networking*, 2019. DOI: 10.1007/S42045-019-00016-8.
- [52] A. Raman, K. Chou, and S. Mastorakis, "A simulation framework for peer-to-peer file sharing in named data networking," *WNS3*, 2019. DOI: 10.1145/3321349.3321357.
- [53] D. S. Sayeed and H. Marco-Gisbert, "Assessing blockchain consensus and security mechanisms against the 51 percent attack," *Applied Sciences*, 2019. DOI: 10.3390/APP9091788.
- [54] D. Tarr, E. Lavoie, A. Meyer, and C. Tschudin, *Secure Scuttlebutt: An Identity-Centric Protocol for Subjective and Decentralized Applications*. ACM, 2019, pp. 1–11. DOI: 10.1145/3357150.3357396.
- [55] *Action Research vs. Design Research*. 2020. DOI: 10.1007/978-3-030-32610-4_8.

- [56] “An introduction to indigenous research ethics,” *Indigenous Research Ethics: Claiming Research Sovereignty Beyond Deficit and the Colonial Legacy (Advances in Research Ethics and Integrity)*, vol. 6, pp. 1–15, 2020. DOI: <https://doi.org/10.1108/S2398-601820200000006001>.
- [57] S. Carroll, I. Garba, O. L. Figueroa-Rodríguez, *et al.*, “The care principles for indigenous data governance,” *Data Sci. J.*, 2020. DOI: 10.5334/DSJ-2020-043.
- [58] S. Hashemipour and M. Ali, “Amazon web services (aws) – an overview of the on-demand cloud computing platform,” *Lecture notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2020. DOI: 10.1007/978-3-030-60036-5_3.
- [59] “Holochain - a framework for distributed applications,” 2020.
- [60] A. Hrga, T. Capuder, and I. P. Žarko, “Demystifying distributed ledger technologies: Limits, challenges, and potentials in the energy sector,” *IEEE Access*, 2020. DOI: 10.1109/ACCESS.2020.3007935.
- [61] K. Janjua, M. A. Shah, A. S. Almogren, H. A. Khattak, C. Maple, and I. Din, “Proactive forensics in iot: Privacy-aware log-preservation architecture in fog-enabled-cloud using holochain and containerization technologies,” *Electronics*, 2020. DOI: 10.3390/ELECTRONICS9071172.
- [62] R. Jung, “Understanding and evolving the rust programming language,” 2020. DOI: 10.22028/D291-31946.
- [63] K. R. Larsen, R. Lukyanenko, R. M. Müller, *et al.*, “Validity in design science research,” *DESRIST*, 2020.
- [64] J. M. Patel, “Introduction to cloud computing and amazon web services (aws),” 2020. DOI: 10.1007/978-1-4842-6576-5_3.
- [65] B. Qin, Y. Chen, Z. Yu, L. Song, and Y. Zhang, “Understanding memory and thread safety practices and issues in real-world rust programs,” *PLDI*, 2020. DOI: 10.1145/3385412.3386036.
- [66] J. Scales, “A design science research approach to closing the gap between the research and practice of project scheduling,” *Systems Research and Behavioral Science*, 2020. DOI: 10.1002/SRES.2743.
- [67] J. Scales, “A design science research approach to closing the gap between the research and practice of project scheduling,” *Systems Research and Behavioral Science*, vol. 37, no. 5, pp. 804–812, 2020. DOI: <https://doi.org/10.1002/sres.2743>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sres.2743>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sres.2743>.

- [68] M. Sudwoj, "Rust programming language in the high-performance computing environment," 2020. DOI: 10.3929/ETHZ-B-000474922.
- [69] C. Tang, J. M. Plasek, Y. Zhu, and Y. Huang, "Data sovereigns for the world economy," *Humanities and Social Sciences Communications*, vol. 7, 1 2020. DOI: 10.1057/s41599-020-00664-y.
- [70] K. Wahlstrom, A. U. Haq, and O. Burmeister, "Privacy by design: A holochain exploration," *Australas. J. Inf. Syst.*, 2020. DOI: 10.3127/AJIS.V24I0.2801.
- [71] M. F. Zia, M. Benbouzid, E. Elbouchikhi, S. Muyeen, K. Techato, and J. Guerrero, "Microgrid transactive energy: Review, architectures, distributed ledger technologies, and market analysis," *IEEE Access*, 2020. DOI: 10.1109/ACCESS.2020.2968402.
- [72] C. Antal, T. Cioara, I. Anghel, M. Antal, and I. Salomie, "Distributed ledger technology review and decentralized applications development guidelines," *Future Internet*, 2021. DOI: 10.3390/FI13030062.
- [73] J. A. Fraire and E. L. Gasparini, "Centralized and decentralized routing solutions for present and future space information networks," *IEEE Network*, 2021. DOI: 10.1109/MNET.011.2100102.
- [74] F. Fritsch, J. Emmett, E. Friedman, *et al.*, "Challenges and approaches to scaling the global commons," *Frontiers in Blockchain*, 2021. DOI: 10.3389/FBLOC.2021.578721.
- [75] K. R. Fulton, A. Chan, D. Votipka, M. Hicks, and M. L. Mazurek, "Benefits and drawbacks of adopting a secure programming language: Rust as a case study," *SOUPS @ USENIX Security Symposium*, 2021.
- [76] J. García-Hernández, L. G. Marín-Collazos, G. Jiménez-Estévez, and P. Mendoza-Araya, "Distributed ledger technologies based microgrid energy management using iota tangle," *CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies*, 2021. DOI: 10.1109/CHILECON54041.2021.9702926.
- [77] "Indigenous and tribal peoples data governance in health research: A systematic review.," *International Journal of Environmental Research and Public Health*, vol. 18, no. 19, pp. 10318–, 2021. DOI: 10.3390/IJERPH181910318.
- [78] S. Kang, A. Eryilmaz, and C. Joo, "Comparison of decentralized and centralized update paradigms for remote tracking of distributed dynamic sources," *IEEE Conference on Computer Communications*, 2021. DOI: 10.1109/INFOCOM42981.2021.9488777.
- [79] M. Leshkowitz, O. Benattasse, O. Wertheim, and O. Rottenstreich, "Scalable blockchain execution via parallel block validation," *Annales Des Télécommunications*, pp. 1–16, 2021. DOI: 10.1007/S12243-021-00857-9.

- [80] S.-J. Moon, S.-B. Kang, and B.-J. Park, "A study on a distributed data fabric-based platform in a multi-cloud environment," *The International Journal of Advanced Culture Technology*, 2021. DOI: 10.17703/IJACT.2021.9.3.321.
- [81] Mozilla, *Mozilla welcomes the Rust Foundation*, May 2021. [Online]. Available: <https://blog.mozilla.org/en/mozilla/mozilla-welcomes-the-rust-foundation/>.
- [82] S. Neela, Y. Neyyala, V. Pendem, K. Peryala, and V. V. Kumar, "Cloud computing based learning web application through amazon web services," *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2021. DOI: 10.1109/ICACCS51430.2021.9441974.
- [83] E. K. Quigan, J. Gaffney, and R. Si'ilata, "Ēhara tāku toa i te toa takitahi, engari he toa takitini: The power of a collective," 2021. DOI: 10.1080/1177083X.2021.1920434.
- [84] T. K. Raraunga, "Iwi data needs paper," 2021. [Online]. Available: https://www.kahuiraraunga.io/_files/ugd/b8e45c_4ecef8047ab4162a3ff07468af5a27d.pdf.
- [85] J. Reeves, G. Treharne, R. T. (Arawa), *et al.*, "Understanding the data-sharing debate in the context of aotearoa/new zealand: A narrative review on the perspectives of funders, publishers/journals, researchers, participants and māori collectives," *K?tuitui: New Zealand Journal of Social Sciences Online*, 2021. DOI: 10.1080/1177083X.2021.1922465.
- [86] C. Robinson, T. Kong, R. Coates, *et al.*, "Caring for indigenous data to evaluate the benefits of indigenous environmental programs," *Environmental Management*, 2021. DOI: 10.1007/s00267-021-01485-8.
- [87] H. Singh, "Security in amazon web services," 2021. DOI: 10.1007/978-1-4842-6222-1_3.
- [88] D. Sordi and J. Osvaldo, "Design science research method," *Research Papers in Economics*, 2021. DOI: 10.1007/978-3-030-82156-2_5.
- [89] A. Aftab, C. Chrysostomou, H. K. Qureshi, and S. Rehman, "Holo-block chain: A hybrid approach for secured iot healthcare ecosystem," *2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2022. DOI: 10.1109/WIMOB55322.2022.9941553.
- [90] "An overview of cloud computing," *International Journal of Advanced Research in Science, Communication and Technology*, 2022. DOI: 10.48175/ijarsct-2899.
- [91] B. S. Anupama, N. R. Sunitha, B. S. Anupama, and N. R. Sunitha, "Analysis of the consensus protocols used in blockchain networks – an overview," 2022. DOI: 10.1109/ICDSIS55133.2022.9915929.

- [92] “Assessing organizational health-analytics readiness: Artifacts based on elaborated action design method,” *Journal of Enterprise Information Management*, vol. 36, no. 1, pp. 123–150, 2022. DOI: 10.1108/jeim-10-2020-0422.
- [93] E. Blanchard, “Client-side hashing for efficient typo-tolerant password checkers,” *International journal of systems and software security and protection*, 2022. DOI: 10.4018/ijsssp.302622.
- [94] J. Bowen and A. Hinze, “Participatory data design: Managing data sovereignty in iot solutions,” *Interacting with computers*, 2022. DOI: 10.1093/IWC/IWAC031.
- [95] “Data sovereignty in community-based environmental monitoring: Toward equitable environmental data governance,” *Bioscience*, vol. 72, no. 8, pp. 714–717, 2022. DOI: 10.1093/biosci/biac048.
- [96] *Design Science Research Methodology and Its Application to Developing a New Timetabling Algorithm*, 2022. DOI: 10.1109/CyberneticsCom55287.2022.9865661.
- [97] C. Feng, B. Liu, K. Yu, S. Goudos, and S. Wan, “Blockchain-empowered decentralized horizontal federated learning for 5g-enabled uavs,” *IEEE Transactions on Industrial Informatics*, 2022. DOI: 10.1109/TII.2021.3116132.
- [98] *Maintaining Control over Distributed Data Through a Data Sovereignty Model*, 2022, pp. 1–7. DOI: 10.1109/ICITDA55840.2022.9971218.
- [99] *Holochains for distributed data integrity*, Oct. 2022. [Online]. Available: <http://ceptr.org/projects/holochain>.
- [100] Y. S. Kiyak, A. Poor, I. İ. Budakoğlu, and Ö. Coşkun, “Holochain: A novel technology without scalability bottlenecks of blockchain for secure data exchange in health professions education,” 2022. DOI: 10.1007/S44217-022-00013-Y.
- [101] D. Kutz, B. A. Cumbie, and M. T. Mullarkey, “Incorporating the student perspective in designing a virtual team classroom environment: An elaborated action design science research approach,” *Journal of Research in Innovative Teaching and Learning*, 2022. DOI: 10.1108/JRIT-02-2022-0007.
- [102] I. Luchnikov, O. E. Tatarkin, and A. Fedorov, “High-performance state-vector emulator of a gate-based quantum processor implemented in the rust programming language,” 2022.
- [103] K. Mannell and E. Smith, “Alternative social media and the complexities of a more participatory culture: A view from scuttlebutt,” *Social media and society*, vol. 8, no. 3, pp. 205 630 512 211 224–205 630 512 211 224, 2022. DOI: 10.1177/20563051221122448.
- [104] G. C. Oliver, S. Lilley, J. Cranefield, and M. Lewellen, “Implementing indigenous data sovereignty: Insights from legislative reform in aotearoa new zealand,” *Proceedings of the Association for Information Science and Technology*, 2022. DOI: 10.1002/PRA2.655.

- [105] A. Poor, “Data sovereignty in action: Designing, building and implementing a radically distributed health information system in aotearoa new zealand,” Ph.D. dissertation, Auckland University of Technology, 2022.
- [106] T. K. Raraunga, “Māori data sovereignty and offshoring māori data,” 2022. [Online]. Available: https://www.kahuiraraunga.io/_files/ugd/b8e45c_c035c550c8244c70a1025cd90a97.pdf.
- [107] S. Rodríguez, “Unit testing,” in 2022. DOI: 10.1007/978-1-4842-8698-2_9.
- [108] I. Sabek, K. Vaidya, D. Horn, A. Kipf, M. Mitzenmacher, and T. Kraska, “Can learned models replace hash functions?” *Proceedings of the VLDB Endowment*, 2022. DOI: 10.14778/3570690.3570702.
- [109] R. Soltani, M. Zaman, R. Joshi, and S. Sampalli, “Distributed ledger technologies and their applications: A review,” *Applied Sciences*, 2022. DOI: 10.3390/APP12157898.
- [110] K.-L. Tan, C.-H. Chi, and K.-Y. Lam, “Analysis of digital sovereignty and identity: From digitization to digitalization,” *arXiv.org*, vol. abs/2202.10069, 2022.
- [111] *The Premises of Design Research*. 2022. DOI: 10.4324/9781003089728-11.
- [112] J. Wang, Z. Xu, X. Wang, and J. Lu, “A comparative research on usability and user experience of user interface design software,” *International Journal of Advanced Computer Science and Applications*, 2022. DOI: 10.14569/ijacsa.2022.0130804.
- [113] S. Zaman, M. R. A. Khandaker, R. T. Khan, F. Tariq, and K.-K. Wong, “Thinking out of the blocks: Holochain for distributed security in iot healthcare,” 2022. DOI: 10.1109/ACCESS.2022.3163580.
- [114] S. Zhu, Z. Zhang, B. Qin, A. Xiong, and L. Song, “Learning and programming challenges of rust: A mixed-methods study,” *International Conference on Software Engineering*, 2022. DOI: 10.1145/3510003.3510164.
- [115] “A study on decentralized web hosting using peer-to-peer architecture,” 2, vol. 3, no. 2, pp. 26–29, 2023. DOI: 10.46632/daai/3/2/6.
- [116] P. T. Brown, D. Wilson, K. West, *et al.*, “Māori algorithmic sovereignty: Idea, principles, and use,” 2023. arXiv: 2311.15473 [cs.CY].
- [117] “Cloud computing: Applications, challenges and open issues,” *arXiv.org*, 2023. DOI: 10.48550/arXiv.2305.17454.
- [118] “Decentralised or centralised management of data and products: Influence on revenue-generating processes,” *International Journal of Management and Decision Making*, vol. 22, no. 1, pp. 74–74, 2023. DOI: 10.1504/ijmdm.2023.127687.
- [119] “Docs — sst.” (Jun. 22, 2023), [Online]. Available: <https://docs.sst.dev/>.

- [120] A. L. Dogan and D. Wood, ““do you collect data to give to the university or do you do the work to benefit people?”: Indigenous data sovereignty in environmental contexts,” *The Compass*, 2023. DOI: 10.1145/3588001.3609368.
- [121] K. Ferdowsi, “The usability of advanced type systems: Rust as a case study,” *ArXiv*, 2023. DOI: 10.48550/ARXIV.2301.02308.
- [122] R. Göttlich, “Testing,” in 2023. DOI: 10.1007/978-1-4842-9234-1_12.
- [123] D. Hardin, “Hardware/software co-assurance for the rust programming language applied to zero trust architecture development,” *ACM SIGAda Ada Letters*, 2023. DOI: 10.1145/3591335.3591340.
- [124] *Holochain core concepts: What is holochain?* 2023. [Online]. Available: https://developer.holochain.org/concepts/1_the_basics/.
- [125] Y. S. Kzyak, “Blockchain and holochain in medical education from planetary health and climate change perspectives,” *Revista Española de Educación Médica*, 2023. DOI: 10.6018/EDUMED.560581.
- [126] T. Kukutai, “Indigenous data sovereignty-a new take on an old theme.,” *Science*, vol. 382, no. 6674, ead14664–ead14664, 2023. DOI: 10.1126/science.ad14664.
- [127] A. Kumar, T. Mehrotra, and G. K. Rajput, *A review on double spending problem in blockchain*, 2023, pp. 881–889. DOI: 10.1109/CISES58720.2023.10183579.
- [128] M. Liao, J. Wang, C. Chen, and S. S. Sundar, “Less vigilant in the mobile era? a comparison of information processing on mobile phones and personal computers,” 2023. DOI: 10.1177/14614448231209475.
- [129] H. Ltd, *Holo Hosting for P2P apps — Powered by Holochain*, Dec. 2023. [Online]. Available: <https://holo.host/>.
- [130] M. Luczak-Rösch, M. Galster, and K. Shedlock, “The veracity grand challenge in computing: A perspective from aotearoa new zealand,” *Communications of the ACM*, 2023. DOI: 10.1145/3589154.
- [131] S. Lyu and A. Rzeznik, “Welcome to the world of rust,” in *Apress eBooks*, 2023, pp. 1–8. DOI: 10.1007/978-1-4842-9331-7_1.
- [132] J. Prehn and M. Walter, “Indigenous data sovereignty and social work in australia,” *Australian Social Work*, 2023. DOI: 10.1080/0312407X.2023.2186256.
- [133] T. K. Raraunga, “Māori data governance model,” 2023. [Online]. Available: https://www.kahuiraraunga.io/_files/ugd/b8e45c_a5b7af8b688c4cd9b7583775c27da52e.pdf.
- [134] R. Theodore, A. Boulton, and A. Sporle, “Māori linked administrative data,” *International Indigenous Policy Journal*, 2023. DOI: 10.18584/IIPJ.2023.14.1.13412.

- [135] P. Menard and G. J. Bott, “Artificial intelligence misuse and concern for information privacy: New construct validation and future directions,” *Information Systems Journal*, 2024. DOI: 10.1111/isj.12544.
- [136] “Āhau — researchh.” (), [Online]. Available: https://ahau.io/nga_taonga.html.
- [137] “Āhau — technology.” (), [Online]. Available: <https://ahau.io/technology.html>.
- [138] “Anchors in hdk::prelude - rust.” (), [Online]. Available: <https://docs.rs/hdk/latest/hdk/prelude/struct.Anchor.html>.
- [139] H.-B. E. Brock Arthur, *Ceptr Revelation*. [Online]. Available: <http://ceptr.org/whitepapers/revelation>.
- [140] *Daisyui — tailwind css components (version 4 update is here)*. [Online]. Available: <https://daisyui.com/>.
- [141] “Get started — requirements.” (), [Online]. Available: <https://developer.holochain.org/get-started/>.
- [142] “Hdk::capability - rust.” (), [Online]. Available: <https://docs.rs/hdk/latest/hdk/capability/index.html>.
- [143] “Hdk::info - rust.” (), [Online]. Available: <https://docs.rs/hdk/latest/hdk/info/index.html>.
- [144] “Hdk::x_salsa20_poly1305 - rust.” (), [Online]. Available: https://docs.rs/hdk/latest/hdk/x_salsa20_poly1305/index.html.
- [145] Holochain, *holochain-proto/holochain.pdf at whitepaper · holochain/holochain-proto*. [Online]. Available: <https://github.com/holochain/holochain-proto/blob/whitepaper/holochain.pdf>.
- [146] “Holochain — development roadmap.” (), [Online]. Available: <https://www.holochain.org/roadmap/>.
- [147] “Holochain — how the holochain framework works.” (), [Online]. Available: <https://www.holochain.org/how-does-it-work/>.
- [148] “Holochain — projects.” (), [Online]. Available: <https://www.holochain.org/projects/>.
- [149] “Holochain — what is holochain?” (), [Online]. Available: <https://www.holochain.org/what-holochain/>.
- [150] Holochain-Open-Dev, *GitHub - holochain-open-dev/profiles: Profile management zome for Holochain hApps with at least a nickname*. [Online]. Available: <https://github.com/holochain-open-dev/profiles>.
- [151] “Jsonplaceholder - free fake rest api.” (), [Online]. Available: <https://jsonplaceholder.typicode.com/>.

- [152] “Moving away from cdk.” (), [Online]. Available: <https://sst.dev/blog/moving-away-from-cdk.html>.
- [153] “Path in hdk::prelude - rust.” (), [Online]. Available: <https://docs.rs/hdk/latest/hdk/prelude/struct.Path.html>.
- [154] “Pepeha.” (), [Online]. Available: <https://www.pepeha.nz/>.
- [155] “Scuttlebutt protocol guide.” (), [Online]. Available: <https://ssbc.github.io/scuttlebutt-protocol-guide/>.
- [156] *Tailwind css - rapidly build modern websites without ever leaving your html.* [Online]. Available: <https://tailwindcss.com/>.
- [157] “Validation: Assuring data integrity.” (), [Online]. Available: https://developer.holochain.org/concepts/7_validation/.